

IEEE

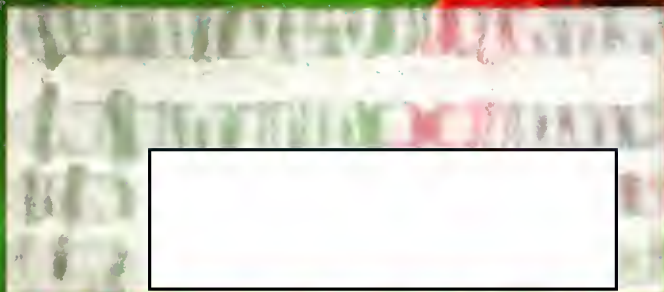
MICRO

APRIL 1993

Chips, Systems, Software, and Applications

Advanced Packaging and Interconnections

Coordinated issue with April 1993 *Computer*
on Multichip Modules



IEEE COMPUTER SOCIETY



THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.



IEEE MICRO

Published by the IEEE Computer Society

Volume 13 Number 2

April 1993

F E A T U R E S

THEME ARTICLES

- 7 Guest Editor's Introduction: Advanced Packaging and Interconnection Technology**
David Misunas
-
- 10 Addressing the Challenges of Advanced Packaging and Interconnections**
Dennis J. Herrell
Realizing high density and high performance at low cost through consortium-based cooperative research
-
- 19 Trends in Low-Cost, High-Performance Substrate Technology**
David H. Carey
Reducing interconnection cost and production time through use of multichip modules
-
- 28 Programmable Multichip Modules**
Surendra Burman and Naveed A. Sherwani
Using one generic layout for implementing several different designs based on user requirements
-
- 36 Interaction of Multichip Module Substrates with High-Density Connectors**
Yakov Belopolsky
Producing fast, cost-effective, good-quality designs, reducing stresses, and improving system reliability with computer modeling and analysis
-
- 46 Amalgams for Improved Electronics Interconnection**
Colin A. MacKay
Replacing traditional lead-based solders with amalgam systems to reduce cost and protect the environment
-
- SPECIAL FEATURES**
-
- 59 A 16-Kbit Θ -Search Associative Memory**
A.R. Hurson and Patrick M. Miller
Introducing a high-performance, high-capacity design that increases the versatility of associative memory in handling variable and long word lengths
-
- 66 Special Report: Virtual Reality in Japan**
David K. Kabaner
Giving users a 3D, interactive "feel" for computer images

Cover image: Bryan F. Peterson, The Stock Market

Cover design: Chris Martin, Design and Direction

Circulation: *IEEE Micro* (ISSN 0272-1732) is published bimonthly by the IEEE Computer Society, PO Box 3014, Los Alamitos, CA 90720-1264; IEEE Computer Society Headquarters, 1730 Massachusetts Ave., NW, Washington, DC 20036-1903; IEEE Headquarters, 345 East 47th St., New York, NY 10017. Annual subscription: \$23 in addition to IEEE Computer Society or any other IEEE society member dues; \$42 for members of other technical organizations. This journal is also available in microfiche form.

Postmaster: Send address changes and undelivered copies to *IEEE Micro*, PO Box 3014, Los Alamitos, CA 90720-1264. Second-class postage is paid at New York, NY, and at additional mailing offices. Canadian GST#125634188.

Copyright and reprint permissions: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US Copyright Law for private use of patrons those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 29 Congress St., Salem, MA 01970. For other copying, reprint, or republication permission, write to Permissions Editor, *IEEE Micro*, PO Box 3014, Los Alamitos, CA 90720-1264. Copyright © 1993 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Printed in USA.

DEPARTMENTS

- 2 In the Mailbag**
- 3 Micro View**
Neural-network chip
- 6 Micro Review**
Death, taxes, and humor
- 74 Micro Law**
A back door to protection?
- 77 Micro Standards**
Standards and innovation
- 79 Software Report**
Micromachines; high-tech odds and ends
- 83 New Products**
Chips and devices; DSP software/hardware; communications; design software
- 87 Product Summary**

Call for Articles, p. 2; Reader Service cards, p. 80A/B; Advertiser/Product Index/ Moving Coupon, p. 88; Computer Society information, cover 3

IEEE Computer Society
PO Box 3014, Los Alamitos, CA 90720-1264
(714) 821-8380

Editorial: Unless otherwise stated, bylined articles and descriptions of products and services reflect the author's or firm's opinion; inclusion in this publication does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society. Send editorial correspondence to *IEEE Micro*, PO Box 3014, Los Alamitos, CA 90720-1264. All submissions are subject to editing for style, clarity, and space considerations.

EDITOR IN CHIEF

Dante Del Corso
*Politecnico di Torino**

ASSOCIATE EDITORS IN CHIEF

K.E. Grosspietsch
GMD

Ashis Khan
Mips Technologies, Inc.

Maurice Yunik
University of Manitoba**

EDITORIAL BOARD

Stephen L. Diamond <i>SunSoft, Inc.</i>	Gilles Privat <i>France Telecom</i>
Joe Hootman <i>University of North Dakota</i>	Ken Sakamura <i>University of Tokyo</i>
Victor K.L. Huang <i>National University of Singapore</i>	John L. Schmalzel <i>University of Texas at San Antonio</i>
David K. Kahaner <i>National Institute of Standards and Technology</i>	Arun K. Sood <i>George Mason University</i>
Hubert D. Kirmann <i>Asea Brown Boveri Research Center</i>	John W. Steadman <i>University of Wyoming</i>
Priscilla Lu <i>AT&T</i>	Richard H. Stern <i>Oblon, Spivak, McClelland, Mauer & Neustadt</i>
Richard Mateosian	Osamu Tomisawa <i>Mitsubishi Electric Corporation</i>
Teresa H. Meng <i>Stanford University</i>	Philip Treleaven <i>University College London</i>
Nadine E. Miner <i>Sandia National Laboratories</i>	Uri Weiser <i>Intel Israel, Ltd.</i>

MAGAZINE OPERATIONS COMMITTEE

James J. Farrell, III (chair)
Valdis Berzins
B. Chandrasekaran
Carl Chang
Manuel d'Abreu
Dante Del Corso
John A.N. Lee
Ted Lewis
Michael J. Quinn
Peter R. Wilson

PUBLICATIONS BOARD

Barry W. Johnson (chair)
James J. Farrell, III
Ronald G. Hoelzeman
Mary Jane Irwin
Ming T. (Mike) Liu
Michael C. Mulder
Theo Pavlidis
David C. Rine
Sallie V. Sheppard
Harold Stone
Kishor Trivedi

STAFF

Marie English
Managing Editor
Dick Price
Staff Editor
H.T. Seaborn
Publisher
Marilyn Potes
Editorial Director, CS Magazines
Jay Simpson
Art Director
Joseph Daigle
Design/Production
John Gill
Membership/Circulation Manager
Heidi Rex
Advertising Manager
Marian Tibayan
Advertising Coordinator

* Submit six copies of all articles and special-issue proposals to Dante Del Corso, Dipartimento di Elettronica, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129 Torino, Italy; phone +39-11-564-4044; Compmail: d.delcorso; Internet: delcorso@polito.it; Bitnet: delcorso@itopoli or

** Maurice Yunik, Dept. of Electrical Engineering, University of Manitoba, Winnipeg, Manitoba R3T 2N2, Canada; phone (204) 474-8517; yunik@eeserv.umanitoba.ca

Author guidelines for article submission are available from the West Coast office (address at left).

Mailbag

(LK: liked; DLK: disliked; LTS: like to see)

February 1992

LK: "Hardware Requirements for Neural Network Pattern Classifiers" and "Experimentation with Hypercube Database Engines."—D.P., Pietermaritzburg, South Africa

June 1992

LTS: Phone numbers of companies in the New Products section.—T.V., Northfield, MN (You will find the names of companies for each new product; the usual contact point is the local representative.—D.D.C.)

LTS: As much as possible on legal issues in software and hardware!—Anonymous (Micro Law is one of our best departments; it will continue.—D.D.C.)

LK: Video; DLK: articles that take too long to read or research; LTS: V.42

bis, F.FAST modem algorithms.—J.T., Kilsyth, Australia

LK: "Associative Processor and Memories: A Survey," and "Cascading Content-Addressable Memories."—S.F., Teheran, Iran

LTS: ... programs in PCB design; details about 1) combination Ethernet boards, 2) power supply design (specially UPS.)—T.K., Damascus, Syria (These issues are better addressed in other IEEE publications.—D.D.C.)

August 1992

LK: TMS320C40 article is very good.—K.W.S., Acton, MA

LK: The article on CMOS technology trends and economics.—B.S., Austin, TX

LTS: Designing RISC microprocessors and FPU/MMU chip set.—S.H.L., Seoul, Korea

LTS: CAD/CAM, CIM/FMS.—P.P.G., Trivandrum, India

LTS: Fuzzy logic chips.—M.M., Tehe-

ran (Something is coming.—D.D.C.)

LK: CMOS technology trends. LTS: on-board computers for Space Shuttle and launch vehicle.—S.O., Trivandrum, India

October 1992

LK: Very good item on piracy law—complete with what IEEE is doing about the mess.—T.P., Sprecilels (Congratulations and questions routed to the Micro Law Editor, Richard Stern.—D.D.C.)

LK: Guest Editor's Introduction; DLK: "Tools to make the engineer's life easier."—T.S., Tokyo

December 1992

LK: Articles on software books; LTS: more articles of software and related books.—R.E.H., Daly City, CA

LTS: E-mail address for comments.—Anonymous (Look on page 1, towards the bottom.—D.D.C.)

Call for Articles

IEEE Micro seeks general interest manuscripts for 1993 and 1994 issues.

Suggested topics include portables, low-voltage systems, wireless transmission systems, speech recognizers, biological computing, workstations, multiprocessing, optical processing, and optical character recognizers.

Submit six copies of papers to either:

Editor-in-Chief Dante Del Corso
Dipartimento di Elettronica
Politecnico di Torino
C.so Duca degli Abruzzi, 24
10129 Torino, Italy
e-mail: delcorso@polito.it

Associate EIC Maurice Yunik
Dept. of Electrical Engineering
University of Manitoba
Winnipeg, Manitoba R3T 2N2, Canada
phone (204) 474-8517
e-mail: yunik@eeserv.ee.umanitoba.ca



For author guidelines, contact Claire Weller, IEEE Computer Society West Coast Office, phone (714) 821-8380 or fax (714) 821-4010.



Ware Myers

Contributing Editor

Market volume drives neural-net technology

How practical is neural network technology, and how has it been put to work? *IEEE Micro* sought some answers to these questions in a recent interview with Federico Faggin, president and chief executive officer of Synaptics Inc. in San Jose, California.

Synaptics was founded in 1986 by Faggin and Carver A. Mead, professor of computer science at the California Institute of Technology, Pasadena. (Mead serves as chair of the company.) Faggin received a doctorate in physics from the University of Padua, Italy, developed silicon-gate technology at Fairchild Semiconductor, and directed microprocessor development at Intel in the early 1970s. Since 1974 he has been a founder and CEO of Zilog, Cygnet Technologies, and Synaptics.

Synaptics developed the first neural-network silicon chip. The I-1000 is used in the Onyx Check Reader, introduced by VeriFone Inc. of Redwood City, California, in June 1992, for use in the banking industry.

Banking originally designed the MICR (Magnetic Ink Character Recognition) line along the bottom of checks to be read by a magnetic reader. The reader operates by sensing the flux change as a single magnetic gap scans the characters. In practice, however, some checks fail to meet specifications on such factors as ink density, check alignment, and so on.

Other checks are folded or crumpled by that well-known wild card—a human being. Where the magnetic reader fails to recognize check characters under these nonstandard circumstances, a human eye easily does so. Similarly, a neural network, mimicking some functions of the eye, can also do so.

Is Synaptics the first company to have a

neural-network silicon chip used in a commercial product?

That is correct. Previously, few neural-network applications had been simulated on conventional computer hardware. This simulation verified the concept, but operated slowly. In hardware a neural network can operate in parallel. In a simulation on one digital processor, of course, computations have to be performed serially.

How do neural networks function?

A single neuron typically combines a large number of inputs with the same number of parameters or weights to produce an output. You can combine neurons to create many layers or strata. The outputs of the neurons in the first layer become the inputs to the second layer, and so on. The values of the parameters associated with a neuron are determined by a process of learning. [For a brief tutorial on neural network basics, see page 35, Dec. 1992, *IEEE Micro*.]

How does your check-character recognition chip work?

There are only 14 different characters in the MICR system, so we have 14 recognition neurons. The chip contains an area image and means to detect and isolate the character it is currently sensing into 400 picture elements. It applies these signals, representing the blackness of each picture element, to 400 multiplier-add circuits in each neuron. Each circuit combines a signal input with an appropriate parameter to produce an output.

The parameters (weights) in each neuron have been set by a process of learning to respond optimally to the pattern of one of the 14 characters. Since the chip processes these 14 times 400 inputs in parallel, recognition is extremely fast.

How do you determine the weight assigned to each circuit?

As many of your readers may already know, neural networks are not programmed by hand in the way conventional digital computers are. They are not even given "rules" in the expert-system sense. Rather, they learn by example, the same way children learn to speak their mother tongue. Children hear words and gradually adapt their neurons to make sense of them and to reproduce them. Similarly, a neural network, in this case, views examples of MICR lines and adjusts the values of its weights when it successfully identifies a character. With each trial the performance of a neural network improves—in our case reaching 99.995 percent or better.

Do your silicon chips learn in this way?

No, not at present. It is one of the goals of the field to reach a structure that continuously learns even after it is deployed in the field, but the technology for doing it is not available at this point.

How do you learn what the weights ought to be then?

The learning is carried out in a simulation of the chip in a conventional computer. Of course, the learning process is fairly slow, because of the limitations of the simulation, as compared to what it would be on a real chip designed for learning. But there is no great rush in the laboratory. Then we cast the parameters determined in this separate learning process in silicon.

How many neurons can you put on a chip?

A better way of expressing the capability of a neural-network chip than giving the number of neurons is to state the number of parameters. A neuron can have many tiny processors, each with a parameter or weight. The I-1000 used in the check reader has approximately 20,000 parameters.

How big is this chip?

It is about 180 by 200 mils, or 4.6 by 5.0 mm.

Are 20,000 parameters the limit of what you can cast on this chip?

No, we can put up to 100,000 parameters on a low-cost chip.

Then your process doesn't limit you to applications the size of the check reader?

No, but we are limited by cost. In this application the whole check reader system has to sell for about \$250.

Some of your chips have now been out in the field for almost a year. How are they holding up?

Pretty much the same as other digital circuits. During our six years of research before we introduced this product, we developed adaptive analog VLSI technology for neural networks. We based it on standard CMOS chip-fabrication processes long employed in high-volume manufacturing. We invented new architectures and circuits, but we avoided exotic processing technologies. In this way we brought to analog technology the advantages of low cost, wide availability, and established reliability that result from the CMOS digital learning curve.

So the hardware is reliable. If a character is quite distorted, does the chip fail to recognize it?

Oh yes. Neural networks are not going to solve problems that are not deterministic. Take a number of humans, put them in a room, give them some borderline cases, and they will not agree on the class that should be assigned to such cases. There will always be gray areas where it is not clear that a distorted pattern is class A or class B.

Does your neural network know when it fails, or does it incorrectly classify something where it should have admitted failure?

We always give a confidence figure. If the figure is low, you know you could have an error. Say you have a letter "O" that looks like a "C" because it is not quite closed. The check reader calls the one with the higher confidence figure.

In this situation a human would look at the context in which the letter appears.

You could introduce context into a neural network application, of course. In the check-reader application, however, there is no context. One character is as likely as another. If the confidence figure for a distorted character is low, users have two choices. They can say they don't know and divert the check to some human process like telephoning the bank. Or they can call it and accept the risk of a mistake. Borderline cases are inherent in this class of recognition problems.

Do neural networks potentially make better distinctions than algorithm- or rule-based systems?

Yes, they do a better job. It is no longer a matter of speculation.

It seems then that you have a reliable chip that makes accurate distinctions at a low cost. I suppose you could extend the present 14-character application to the entire alphabet without much trouble?

It could be done, and character recognition is an area of interest to us. The postal service is an example of a real-world application. Its optical character recognition machines, which are state of the art, successfully sort only about 60 percent of the addresses. Human sorters handle the other 40 percent. If you look at these envelopes, you find that it is sometimes difficult to find the address, or something obstructs a portion of it, or a worn-out ribbon blurs it. With better character recognition, the post office could get better results.

Another example is printed matter

that arrives by fax. The characters, particularly the very small ones, are so corrupted that the recognition rate of rule-based systems is abysmal. People have solved this application with neural networks. In fact, two products on the market use neural networks to get a decent recognition rate after fax corruption.

Can you extend neural networks beyond alphabet character recognition?


We are moving in the direction of object recognition—low-cost, fully integrated vision modules. At first the objects will be very simple ones. As our knowledge increases, they will become more complex.

At some point, maybe five to 10 years from now, it will be possible to have several intelligent electronic eyes at road intersections. They will have to recognize only a few classes, perhaps motorcycles, cars, trucks, police and fire vehicles. With this information the electronic eyes will control the traffic light with much more intelligence than the switches embedded today in the roadway. They should cost less than embedded wires, too.

I envision, perhaps 20 years off, one of these eyes in every room. As people go in and out of the room, it would turn on and off lights, heat, air conditioning, and burglar alarms. The market potential for a low-cost electronic eye is enormous.

The check reader, post office recognition, and electronic eyes are big applications. You seem to be bewitched by volume.

Volume is an absolutely essential requirement to drive the technology. Learning curve or experience curve theory is well documented. It shows that every time you double volume, you lower cost. An 80-percent learning curve, for example, means that every time you double your cumulative volume, you lower your cost to 80 percent of what it used to be.



What's #1 in the hearts of its readers?

IEEE Micro!!

How does it do it?

With quality articles!

Who keeps the quality high?

The article reviewers!

Become a reviewer and join the #1 team.

Want to help keep *IEEE Micro* #1? Editor-in-Chief Dante Del Corso seeks more technical reviewers—people interested in seeing that the articles published in *IEEE Micro* continue to be of the highest quality. The technical review process is a crucial step in providing readers with correct and timely information so they can keep up with their ever-changing profession.

Send your professional information directly to:

Dante Del Corso
Dipartimento di Elettronica
Politecnico di Torino
C.so Duca degli Abruzzi, 24,
10129 Torino, Italy

When you drop your costs, you can apply the resources supplied by the savings to getting a better product. That, in turn, drives the volume still higher. So, the more volume you get, the more energy you can put into improving the product.

The idea of the learning curve can be expressed in the negative, too. If you double the volume, you make perhaps 20-percent fewer mistakes. The learning curve basically reflects a cycle. Every time you go through it you learn something. You reduce your costs and errors.

At first it is like riding an exponential curve, but eventually exponential curves bend over.

Yes, a particular curve bends over when it hits some limit—the limits of physics, or the methods you are using, or even more important, a market limit. If you have enough market, however, people will put money into alternative technologies. They will push physical limits, for example, beyond what they were with the older technology. If the market begins to saturate, however, money to invest in creating new technology decreases. Then you begin to see saturation.

Do some predictors of the future overlook that?

Yes, in technology you have to have volume to support the learning curve that allows you to pursue this effect. Otherwise, you will be overtaken by faster running technologies. In our case digital technology is growing very rapidly. It has tremendous momentum. For this class of problems, we are proposing adaptive analog neural-network technology. Well, we better have volume to back it up and to learn as fast as digital technology is learning, because otherwise we are not going to make it.

One thing I know for sure. A necessary condition, not sufficient in itself for success, but absolutely necessary, is high volume.

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 186 Medium 187 High 188

Micro Review

Richard Mateosian

2919 Forest Avenue

Berkeley, CA

94705-1310

(510) 540-7745

Death, taxes, and humor

The Man Who Knew Infinity—A Life of the Genius Ramanujan, Robert Kanigel (Scribners, New York, 1991, 447 pp.; \$27.95)

S. Ramanujan was the most highly original mathematician of the 20th century. He was a southern Indian Brahmin of extremely modest means and limited education, who attributed his insights to the goddess Namagiri. Barely understood in his own country and desperate for peers and patronage, he risked expulsion from his caste by traveling overseas to Cambridge, England, the citadel of mathematics. There he subjected his untutored ideas to the rigorous scrutiny of G.H. Hardy and J.E. Littlewood, England's finest mathematicians.

Ramanujan's five-year sojourn in England encompassed the entire period of World War I. This probably led to his early death. His vegetarian diet kept him from the camaraderie of communal dining. The war took the attention of his patrons, leaving him further isolated and depressed. Wartime rationing, interruption of food shipments from India, and inadequate sunlight led to malnutrition, then tuberculosis. For two years the danger of ocean travel kept him from going home, where he might have recovered his health. In 1918 he returned to India as a newly elected Fellow of the Royal Society and a national hero. He died in 1920 at the age of 32.

Kanigel weaves many themes into this biography. He elucidates in lay terms some of the more accessible parts of Ramanujan's work. This helps us understand why Ramanujan's greatness was so hard to recognize and gives us insight into the unevenness of his self-directed mathematical development.

In the almost three quarters of a century since Ramanujan's death, the world has not yet understood all that he left behind. His terse, eccen-

tric notebooks have appeared in facsimile, but no one has yet understood them thoroughly. Their contents are a mixture of lengthy elementary calculations, pretty formulas, and deep theorems, with no indication which are which. In 1976 an American, George Andrews, found "the lost notebook," 140 pages of Ramanujan's work from the last year of his life, lying forgotten in the Trinity College Wren Library.

Since Ramanujan was little concerned with rigorous proof, his notebooks are a mixture of correct and incorrect speculation. One of his assertions, a highly unintuitive number-theoretic formula called the tau conjecture, had attracted the attention of mathematicians while Ramanujan was still alive, but no one was able to prove it until 1974. The proof of the tau conjecture added significantly to his already great reputation and sparked a revival of interest in his work.

Kanigel includes a sketch of G.H. Hardy's origins and education. Hardy was Ramanujan's principal sponsor in England, and this sketch helps us understand the not entirely satisfactory relationship between these two men.

A fascinating theme of the biography is the contrast of cultures between Hardy's reserved, secularly intellectual Cambridge and Ramanujan's warm, spiritually intellectual southern Indian Brahmin world. I found the account of Ramanujan's life in India extremely illuminating. Despite great differences, I found much to identify with in Ramanujan's experience.

Ramanujan's great confidence in his own work was balanced at another level by enormously low self-esteem. At the age of nine, he came in second by one point on a mathematics examination. He felt powerfully ashamed at the fact that everyone knew he had not scored highest;

continued on p. 81



Advanced Packaging and Interconnection Technology

David Misunas

*Microelectronics and
Computer Technology
Corporation*

The requirement for ever-more-powerful computing capabilities addressing the growing needs of low-cost, high-performance applications is forcing significant changes in the computer industry. In many cases we are reaching the economic limit on the additional computational power that can be squeezed into devices themselves. While new materials hold the promise of improving device performance, they also leave unresolved issues of density and cost. Parallel processing and other advanced computer architectures demonstrate improved performance; however, such solutions will not address the majority of applications expected for computers within the next decade.

The most significant advances in the near future will come from improvements in electronic packaging and interconnection technologies. Devices will be assembled to take maximum advantage of chip performance in ways that are compact, lightweight, and cost-effective.

Only recently have advanced packaging and interconnection technologies begun to play their increasingly important role in electronic systems design and manufacturing. Today's emphasis on portability in electronic products and the movement of low-end commercial systems into the over-100-MHz performance range require a new level of sophistication and expertise in packaging technologies. As a result, our prior treatment of packaging as an unglamorous, "late-end" re-

quirement in the design process is no longer viable. We must ensure that consideration of packaging and interconnection requirements and technologies becomes an early and integral part of the overall system design.

We need new substrate technologies, including multichip modules and advanced printed circuit boards, to support the smaller sizes, higher operating speeds, and greater power and heat dissipation requirements of new systems. To attach increasingly smaller packages to these substrates, we need sophisticated assembly, bonding, test, and repair technologies. These packages require lead pitches (center-to-center lead measurements) as low as 4 mils (100 microns). In addition, unpackaged bare die are often directly attached to substrates, either with wire bonds connecting the die to the substrate or with direct connections from die pads to substrate pads using adhesive or metallurgical flip-chip technologies.

Power, noise, and thermal performance issues all play important roles in the new packaging environment. For example, individual die may dissipate tens of watts in small packages. The product design process must incorporate plans to remove this heat and provide clean power to the system. We also need new materials developments to provide organic and inorganic coatings, either separately, or combined in several layers, for environmental protection of bare die mounted on substrates.

Glossary

You may find the following in several of the articles in this issue.

Amalgam: A nonequilibrium, mechanically alloyed material formed at or near room temperature between a liquid metal and a powder

Anisotropic: Exhibiting properties with different values when measured along axes in different directions, as in a crystal

Bonding: Process of establishing electrical connection between the electrical pads on a chip and the conductors that link the chip to the circuit board (inner lead bonding) and between those conductors and the pads on the circuit board (outer lead bonding)

Chip on glass: Method of attaching electronic components to a glass substrate, as on the periphery or back side of a computer screen or calculator display panel

CTE: Coefficient of thermal expansion

Eutectic: An alloy or solution having the lowest melting point possible

Few-chip packaging (FCP): Housing several chips together, for example, a microprocessor and cache, in one package

Flip-chip bonding: Process that attaches chips to substrates by placing the chip face down on the circuit board or other substrate and directly connecting electrical bonding pads on the chip to pads on the board, either with solder or electrically conductive adhesive

Known-good die (KGD): Tested and burned-in chip (die) that both the supplier and the substrate assembler are confident will work properly

Multichip modules (MCMs): Packaging technique characterized by multiple chips attached to a high-density substrate using fine-line wiring technologies and high device-to-board-area ratios

Panelization: Manufacturing many items as one large item

Poisson's ratio: A ratio of the shear yield stress versus the tension yield stress

QTAI: Quick-turnaround interconnection

Quad flat pack (QFP): Square ceramic or plastic chip package with leads projecting down and away from all four sides

Single-chip packaging (SCP): Housing just one chip in each individual package

Surface-mount technology (SMT): Method of board assembly in which component parts are mounted onto, rather than into, circuit boards or other substrates

Tape-automated bonding (TAB): Process by which bare chips are attached to electrical connections in a polymer tape (inner lead bonding), the other ends of which are subsequently attached to a board on a substrate (outer lead bonding)

Wire bonding: Widely used method of fastening individual wires between bonding pads on a chip and pads in a package or on a substrate

Other new compliant materials and assemblies are necessary to accommodate the differing coefficients of expansion of silicon and low-cost interconnection substrates such as PCBs. And, the availability of tested and burned-in bare die for assemblies is causing semiconductor manufacturers to revamp the way they do business, including significant new capital and resource investments.

Most packaging technologies such as multichip modules, tape-automated bonding (TAB), or flip-chip assembly are not new; high-end military and commercial applications have used these technologies for years. However, the current emphasis on commercial portable systems with their attendant requirements for low cost and high volume is driving these advanced technologies to be competitive with existing low-end solutions. In addition, the threat of competition from new and innovative packaging approaches prompts the extension of existing technologies to meet the challenge. As an example, the challenge of high-density, thin-film multichip module technology with 2-mil (50-micron) lines and spaces has forced

extensions to conventional PCB technologies. Such advanced PCB technologies can now provide similar packaging densities with 2- and 3-mil lines and spaces.

We are, however, a long way from fully realizing the promise of advanced packaging and interconnection. Many of the necessary supporting technologies, such as design tools, are only now in development. Future tools will ultimately permit sophisticated trade-off analysis of packaging choices in the design process. For example, they will help us understand the implications for cost, size, and thermal performance of a design implementation using 4-mil pitch TAB technology versus that of a flip-chip implementation. We will see such tools become readily available and increasingly valuable over the next few years as designers gain expertise in balancing the ever-widening range of alternatives available to them.

Perhaps the most critical issue currently blocking widespread use of the new advanced technologies is the infrastructure to support low-cost, high-density, high-performance packaging and interconnection design implementations, which

is still in its infancy. Materials and substrate manufacturers, semiconductor manufacturers, equipment vendors, and design houses are working toward agreement on several significant issues. Critical issues include industry standards, design methodologies, compatible/integratable components and processes, proof of performance and reliability, cost-effective assembly and test methodologies, semiconductor availability in the necessary format and quality, and the ever-complex issue of market demand.

Clearly, we need long-term investment in packaging and interconnection technologies to bring forward advanced technologies. Experience in and application of such low-cost, high-performance packaging and interconnection technologies are now becoming issues of national competitiveness and are attracting the attention of government agencies and national consortia.

THIS ISSUE FEATURES ARTICLES that examine several of these packaging and interconnection requirements from a technology perspective. Herrell provides an overview of technologies and key issues facing the industry. He offers a unique perspective on this area as vice president and division director of the Microelectronics and Computer Technology Corporation (MCC) High Value Electronics Division. His



David Misunas is marketing and business development manager for the High Value Electronics Division at MCC. He holds responsibility for the organization of new research projects within the Packaging/Interconnect Technology Program. He founded and served as president of two

start-up companies in the areas of data communications processors and high-performance computers and has held the position of assistant vice president of marketing and development for Micom Systems, Inc. He has also conducted research in high-performance processors and dataflow computer architectures at MIT's Laboratory for Computer Science.

Misunas received BS and MS degrees in electrical engineering and computer science from the Massachusetts Institute of Technology. He holds six US and two foreign patents and has authored 14 publications on computer architecture and data communications.

responsibilities include the eight-year-old Packaging/Interconnect Technology Program that develops advanced P/I technologies in a consortium of major computer, telecommunications, and aerospace companies.

Carey explores the increasingly important area of interconnection substrate technology, examining recent trends and developments in low-cost, high-performance substrates. He compares advances in PCB technology in response to industry requirements with progress in realizing competitive low-cost thin-film substrates.

Burman and Sherwani further consider the subject of thin-film substrates in the third article on developments in programmable multichip modules. They discuss rapid customization of high-volume, and, thus, potentially low-cost, high-yield, generic interconnection substrates.

The next two articles explore several important supporting technologies. Belopolsky examines advances in connector technology necessitated by the new high-density substrates. MacKay considers amalgams as an attachment alternative to the familiar tin/lead solder, offering the potential for a lower cost and more environmentally benign process.

These five articles barely scratch the surface of the complex issues developing in advanced packaging and interconnection technology. However, they give us a taste of the trends in the area and the promise of lower cost, higher performance solutions to come. ■

Address questions concerning this special issue to David Misunas, Microelectronics and Computer Technology Corporation, 12100 Technology Blvd., Austin, TX 78727-6298; misunas@mcc.com.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 150

Medium 151

High 152



Addressing the Challenges of Advanced Packaging and Interconnection

Packaging and interconnection technology is undergoing significant changes to meet the rapidly evolving requirements of portable electronics products. The need for high density and high performance at low cost demands sophisticated developments in technology. We can meet future portable equipment packaging requirements only through advanced concepts presently in development, including multichip modules, tape-automated bonding, and flip-chip assembly. Supporting technologies such as adhesive assembly, thermal management, and design tools must also make attendant advances. Consortium-based cooperative research and development of technologies addresses these needs while also developing the essential vendor infrastructure.

Dennis J. Herrell

*Microelectronic and
Computer Technology
Corporation*

Electronic packaging is presently undergoing exciting changes, driven primarily by the need for more cost-effective miniaturized packaging of the evermore powerful integrated circuit. Miniaturization forces us to pack more function and performance into a smaller volume. For instance, we see mainframe performance going into workstations, workstation performance into PCs, and PC performance into palmtop devices.

The most dramatic miniaturization challenges come from the demands for increased performance and functionality in portable systems. Included are such functions as computation, telecommunications, and sound and image capture and display. Portable products and applications demand significant advances in the electronic packaging technologies. We must improve the way we design and manufacture chips to meet the needs for higher performance with lower power for longer battery life. These changes must lead as well to acceptable, simple cooling techniques.

These demands mean that chips must run with lower voltages, which requires better screening and closer attention to the electronic noise environment of the signals and power supplies. The requirements of mixed-signal systems will exacer-

berate these noise problems. For example, RF circuitry may be called upon to share the same enclosure with digital and, perhaps, low-frequency analog circuits. We must devise very dense methods of packaging the chips to house the function in a very small volume. This drive to smaller volume, with more complex functionality and higher performance, will severely test our ingenuity in noise management and cooling.

Miniaturization challenges us to develop packaging approaches that achieve high effective density for the chips within the system. For typical systems today that use single-chip packaging, the area of silicon usually occupies less than 10 percent of the total area of the electronic boards. We need to find inexpensive ways of increasing this silicon coverage to greater than 50 percent. Consequently, we need to get the chips closer together, both in the X-Y plane and by stacking them vertically. We must make technology choices for this miniaturization that are equally or less costly than current packaging methods which primarily involve single-chip packaging on printed circuit boards (PCBs).

Miniaturization significantly aids and simplifies our progress toward higher performance. Getting chips closer together reduces the parasitic

impedances associated with the interconnection. Since much of the off-chip speed penalty comes from the capacitance of the interconnection (switching speed governed by the time constant "RC"), the circuits operate faster when placed in a miniaturized packaging and interconnection environment. Also, as the system becomes smaller, net lengths become shorter compared to the characteristic wavelengths associated with operation of the system clock. Consequently, transmission line effects are not encountered until higher speeds, thereby simplifying some of the layout headaches of high performance systems.

Whereas in the past, consumer products benefited from improvements in "high-end" systems, now "low-end" consumer products more often create innovations that eventually find their way into the more advanced systems. We have conventionally classified many of the portable electronic systems currently demanding improvements in chip and packaging technologies as consumer products. These include camcorders, portable audiotape machines, and cameras. To this rank we can now add cellular phones, organizers, and palmtop computers. These products are in the vanguard of cost-effective, miniaturized electronics packaging technology. As these advances prove themselves in the marketplace, they will soon find their way into the high-end, nonportable systems.

High-density substrates

In addition to the miniaturization of the single-chip package (SCP), another important change sweeping the industry is the multichip module (MCM). As we seek better methods of putting chips together into electronic systems, we are finding cost-effective ways of placing a number of chips together in the same protective package rather than having one chip per package. Higher chip I/O counts and closer off-chip lead spacing, either peripherally or with an area array of I/O, challenge the technology of the high-density interconnections used in MCMs. PCB technology will continue as the technology of choice, not only for the substrate upon which we assemble the complete electronic system using SCPs, MCMs, and discrete chips, but also for the needed interconnection between the chips within the MCM.

Figure 1 shows a PCB layout of an MCM designed for a workstation microprocessor and memory application that achieves high performance and high density at relatively low cost. This design, put together by a consortium of companies working at MCC, features interconnection lines and spaces as small as 2 mils (100- μ m pitch). We used it as an application-specific module to assess the state of readiness of MCM design and manufacturing for ~100-MHz workstation applications. This application used fine-line PCB technology primarily because that has shown itself to be the most cost-effective approach.¹

Innovators have developed an alphabet soup of high-density interconnection technologies for MCMs (MCM-C, MCM-



Figure 1. A PCB layout for an MCM.

D, MCM-L, and various combinations) based upon ceramic, thin-film (deposited), and PCB (laminated) techniques, respectively. These approaches have trickled down from the high-end systems that first required MCMs to achieve their performance targets. While very good, the technologies developed for MCMs and interconnections within the MCMs generally cost too much for the burgeoning low-end consumer electronics market. Our industry now stands at a crossroads about which technology to use for MCM interconnections.

As usual, at the start of a new industry or industry sector, component cost and availability compete in a vicious interplay. New product designers will not commit to MCMs because of their high cost, and the cost remains high with a small market. Thin-film and ceramic technologies face this dilemma for their use as MCM interconnections. If the high-volume markets did exist, the costs for ceramic and deposited thin-film technologies would likely compete with those for PCBs. But no large market for these techniques currently is driving their unit cost low enough to make them attractive to low-cost product designers. However, PCB technology does have a large market, thus industry can develop and supply the finer line, higher density PCB versions, initially supported by the large base of more conventional PCB sales.

MCM-L (for laminated) is a fine-line PCB technology. Whereas conventional PCB technologies usually fall in the 10-plus-mil line and space regime, a number of suppliers have extended this technology to the 5-mil line and space level, and some are experimenting with 2-mil lines and spaces. Along with this decrease in line width are reductions in the via drill and pad diameters. In addition, rather than using only plated through-holes that extend completely through the PCB for the vertical interconnection, many suppliers now use blind vias for interlayer connections within the board.

Besides having an ongoing sales volume of conventional

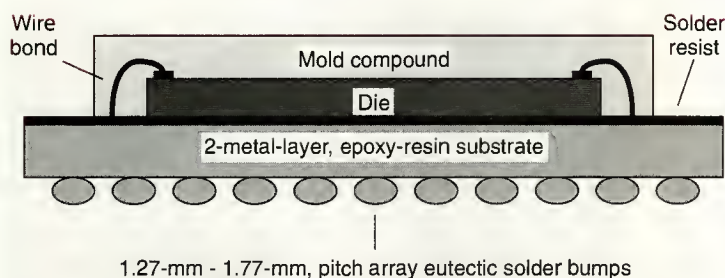


Figure 2. Simplified section of the ball grid array package, also known as the OMPAC.

technology, PCB manufacturers have another significant advantage—they make big boards. Having the tools and equipment for making large-area boards allows them to place many smaller interconnection substrates on a single PCB panel. In much the same way, semiconductor manufacturers place many chip sites on a single wafer. Panelization, or manufacturing many items as one large item, is one of the most important cost-reduction techniques that we have developed. Scaling up and panelizing the thin-film MCM-D approaches remain as challenges. While we developed ceramic substrate technology (MCM-C) with panelization in mind, it remains stalled at the deployment gate because no large-volume, low-cost product market prods it forward.

A cost-effective market incentive for MCM technology requires that we find a way to introduce MCMs into the mainstream of electronic assembly without substantial investment in new factories and equipment. Nor can there be demands for new ways of doing business among the present components of the marketplace—users, vendors, suppliers, systems houses, and semiconductor manufacturers. The current use of few-chip packages (FCPs) provides such an introduction. We use these essentially single-chip packaging approaches to house a few chips rather than a single chip. Quad flat packs (QFPs), pin-grid arrays (PGAs), and the newer solder grid arrays are good examples of package envelopes that we understand well, use pervasively, and can modify to house a few chips rather than one.

Ideal candidates for this modification are applications that require high performance and high I/O connections between neighboring chips. For example, having the microprocessor, I/O, and memory management chips together with a few cache chips, all packaged in an envelope less than 2 inches on a side, certainly seems to be the next step toward increasing performance for workstation and high-end PCs. We can conventionally handle and assemble these few-chip packages into systems presently populated solely with single-chip packages. To get this technology accepted we must find ways to package the few chips into a multichip package for less than or equal to the cost of the parts replaced. That way, we

can improve system performance and significantly reduce system size and cost. This is where the demand is pushing fine-line PCBs, advanced molding, lead forming, and gradual improvements in the plastic SCP technologies.

Figure 2 shows an example of a solder grid array, otherwise known as an OMPAC or ball-grid array. (OMPAC is Motorola's over molded pad array carrier.) Manufacturing for this low-cost, small-footprint package uses standard processes. It has a standard PCB interconnection made with a high-G epoxy resin, industry-standard wire-bond processes and equipment, and a relatively low-cost molding process. High-G, or high glass transition

temperature (T_g), is important for the process windows on subsequent wire bonding, over molding, and attachment. Solder balls (60:40 Sn:Pb) are placed using a template and reflowed in this process.

Until sales of the PCB-based FCP take off, MCM-D and MCM-C apparently will sit on the sidelines. However, further up the cost and performance scale from the new consumer market (systems costing less than \$1,000 to \$2,000) will be system requirements for performance and density that improvements in MCM-L cannot satisfy as yet. There are increasing line and via density requirements beyond which MCM-L cannot cost-effectively reach (typical applications requiring >300 inch²/inch² of interconnection or >400 vias/inch²). A number of companies are seeking to reduce MCM-D costs.

Once again, one of the main cost reductions can come about by increasing the panel size and thereby the "number up" of interconnection substrates for MCMs. ("Number up" is the number of components per workpiece, in this case the number of substrates per panel.) Our low-cost interconnection project focuses on these developments. Process improvements that require fewer steps promise a low-cost approach for MCM-D and the potential for realizing MCM-D manufacturing costs comparable to current PCB costs. Such process improvements include using photopolymers and inexpensive process steps like chemomechanical polishing for planarization and plating for metal deposition as opposed to vacuum deposition. We can also borrow such methods as lamination and rapid laser process via formation from the PCB world.

Other techniques we are developing include exploring ways to introduce high-density interconnection at low cost. One promising approach takes a page out of the semiconductor book. In a process analogous to gate arrays, we have developed an array of interconnections that can easily be personalized for specific applications. This quick-turnaround interconnection (QTAI) design concept allows us to fabricate the vast majority of a high-density interconnection ahead of time, no matter what the application.² This technique applies equally well to MCM-L and MCM-C approaches, although it

looks most promising for MCM-D, which was the technology we used to demonstrate QTAI's effectiveness.

With the interconnection array prefabricated (and stockpiled), a few processing steps will let us personalize it to the specific application. These QTAI personalization steps can incorporate conventional low-cost MCM-D processes, such as plating and polishing, or can be "maskless" and flexibly manufactured using laser direct-write processes. We are currently undertaking just such a laser direct-write project with support from the U.S. Defense Advanced Research Projects Agency (DARPA).

The QTAI approach could solve the severe problem of small sample volumes for MCMs facing early adopters of MCM technology. Small, custom orders have relatively large tooling set-up and other nonrecurring expenses. Suppliers must fabricate relatively large lots because of the low yields found during the early stages of new technology development. A company cannot afford to process too small a lot because a poor yield would prevent them from satisfying the order on time. Producing a larger lot to cover yield extremes can lead to higher costs and an oversupply of specific parts. Further, each new design put into an MCM-D fabrication line has its own peculiarities that can harm the already sensitive yields.

QTAI solves these problems because the application-specific personality of a module can come last in the processing. Only a few samples are needed to explore the application-specific features since the turnaround time for the personalization process takes hours and days, rather than the weeks and months of conventional custom MCM-D processing. If suppliers encounter a yield problem in personalization, they can take a few more unpersonalized QTAI substrates from the stockpile and quickly repeat the job. This flexibility enables them to meet their commitments for delivery of the MCM-D product. We have developed QTAI designs that achieve up to 700 inch²/inch² of interconnection density, addressing requirements up to the level of complexity of a supercomputer.

So far, we have shown that the electrical performance of a QTAI solution matches a fully customized approach. The main high-performance detractor for all interconnection technologies exploited for MCMs arises from the resistance of the fine-line interconnections. This remains true whether we are talking about a custom design or a QTAI design. We can achieve both higher performance custom and QTAI approaches through thicker and wider interconnection lines that have the drawback of lower interconnection density per layer. Perhaps the ultimate performance solution will be to reduce the resistance of the interconnection through low temperature operation. For example, copper at 77K has approximately one sixth of its room temperature resistance.

Before we turn our attention to assembly technologies, we need to discuss the issue of known good die (KGD). This problem could become a serious impediment to the rapid

***The natural way to introduce
MCM technology into
mainstream electronic
products will come through
few-chip packaging.***

adoption and pervasive deployment of MCM technology. Known good die is a problem of infrastructure. How can suppliers ensure that the die they are delivering are "good" and that users of the die know that they are good-working to perfection? How do the interacting companies required for the manufacture of MCMs work together to get cost-effective bare or unpackaged die for incorporation into MCMs? The present semiconductor/electronic systems manufacturing dichotomy conveniently partitions the industry into the chip suppliers and the electronic systems assemblers and systems houses. Semiconductor manufacturers make chips on large wafers, test the wafers, and package the good chips in SCPs that they can test, burn in, ship, and warrant to the user. Industry has developed this format over a number of years; the tools, interfaces, and equipment are available to do this well. Bare die presents a problem that needs solution: the MCM assembler needs a cost-effective supply of KGD in some acceptable shipping container.

We believe that the KGD solution primarily involves 1) wafer-level burn-in and 2) single-die carriers so KGD can be shipped and handled much as SCPs are today. To address these concerns, MCC is currently organizing a project (starting January 1993, jointly with Sematech and funded by DARPA, administered by the U.S. Air Force). The KGD consortium project will assess proposed solutions to these problems, test the alternatives, and make recommendations to the whole industry as to the optimum approaches to take.

Assembly and bonding technologies

Packaging assembly technologies must gradually improve before we will see progress in the cost-effective manufacture of electronic systems. Very substantial changes that took many years culminated with the introduction of surface-mount technology (SMT) to replace the earlier through-hole mounting of components. With the equipment in place and the knowledge in hand, now is not the time to start over with something completely new. Instead, we must gradually evolve the assembly technologies. Consequently, we believe that the natural way to introduce MCM technology into mainstream electronic products will come through FCP.

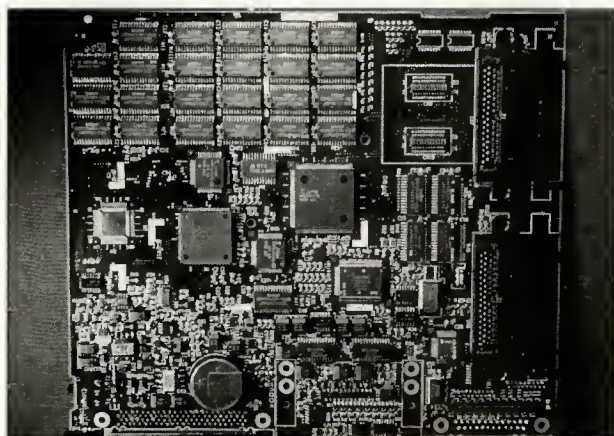


Figure 3. TAB assembly used in the Fujitsu FM-R card notebook computer.

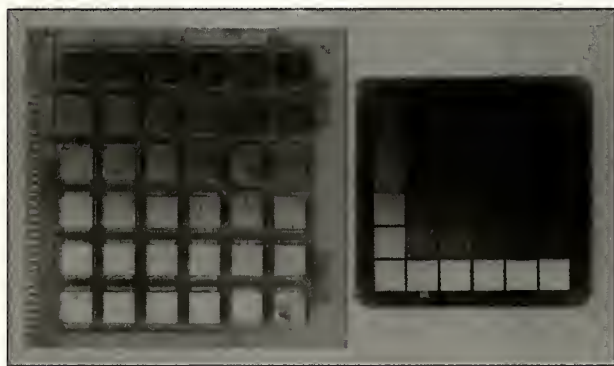


Figure 4. A comparison of the maximum density of TAB (left) and flip-chip (right) technologies.

We need to understand and recognize the limits of present SMT. These limitations arise principally from the increasingly fine lines and connections to packages that we wish to make. A 50-mil pitch between component leads and connections is generally accepted in the industry, with the leading edge somewhere in the range of 20 to 30 mils. That standard depends on how you define the market and whether you are assessing the eastern or western side of the Pacific: Companies located in the United States seem to be well behind other Pacific Rim nations in SMT capabilities. Parenthetically, the demands of high-density consumer electronic requirements, such as for cameras and high-fidelity stereo equipment, have pushed other Pacific Rim manufacturers into aggressive SMT technology faster than market pressures have in North America.

Industry is developing a number of newer technologies

for insertion into consumer products along with the mainline SMT. Tape-automated bonding technology is one of these. In this packaging technique, bare die are bonded to the inner leads of a preformed lead frame that is subsequently formed, placed, and attached to a substrate. TAB allows higher density interconnection than SMT. TAB has long been the province of mass-produced consumer products, especially electronic watches. It now appears that innovators are introducing TAB in selected places where an SCP cannot meet the density and weight demands.

In such a fashion, consumer electronic products (for example, the Fujitsu FM-R card notebook computer) are now shipped with TAB parts incorporating pad spacing as small as 12 mils. See Figure 3 for this TAB assembly. Although claimed to be introduced as a substitute for a QFP to reduce the weight of the overall system below 2 kg, it appears more as an attempt to explore the viability of high lead count TAB—160 I/O—for such applications. Flat panel displays have also witnessed a large growth in TAB technology. Almost completely absent from North American manufacture, this component has current pad spacings as low as 7 mils.

In assembly technology, both for single die and SCPs, flip chip has long been recognized as an ideal goal for chip connection technology. Flip-chip technology involves the attachment of bare die directly onto an interconnection substrate. Either solder balls or conductive adhesives connect pads on the die directly to pads on the interconnection substrate. Figure 4 compares TAB and flip-chip technologies. Each chip with TAB requires a ≥ 50 -mil-wide "picture frame" for the leads and the outer lead bond sites, whereas a flip chip requires ≥ 10 mils to allow for independent chip assembly and rework.

As performance levels reach into the multi-100-MHz regime, recognition of flip chip's superior performance comes primarily from signal and, more importantly, power-fidelity considerations. Other advantages of a singular assembly nature arise from the coarser pad pitch requirements that array connections have over peripheral connections, a particularly important consideration as chip I/O numbers increase.

What then prevents the proliferation of flip-chip technology? There are three interrelated answers: thermomechanical stress, process control, and ease of inspection. Flip-chip technology rigidly connects a device to the high-density interconnection. Very little in the way of flexible structures takes up the stresses and strains of the differential expansion between the component and its substrate. These stresses arise from ambient temperature variations as well as operational temperature gradients (hot chip and cool substrate). Major practitioners of this technology have adopted small temperature extremes for operation (such as very narrow mainframe ambient temperature requirements) and small temperature gradients (for example, flat-panel displays). They also carefully select materials having coefficients of thermal expansion

sion (CTEs) close to that of silicon over the temperature ranges of interest, as with mullite. Fortunately, we are beginning to solve these mechanical problems just when we need the high I/O density that array connections can give us.

New adhesives are providing a breakthrough for flip-chip attachment in general. Anisotropic conductive adhesives, which contain a small distribution of conductive particles in a non-conductive adhesive, as well as adhesives that "get out of the way of critical areas like contacts" are good examples of these developments. Currently, they are replacing solder connections for applications such as flat-panel displays and the attachment of ribbon cables to PCBs. Figure 5 shows an example of an anisotropic conducting adhesive. Note the distribution of conductive particles. We took this sample from a display panel product manufactured by Sharp. Each particle has an approximate diameter of 12 μm and consists of a nickel/gold-plated polymer sphere. When used, the material is squeezed and heated between conductive mating surfaces (usually gold and indium tin oxide—ITO). The spheres make conducting bridges. The resistance of such a contact is less than the trace resistance associated with the ITO lines of the display.

Inserting adhesive films under conventional solder flip-chip connections ameliorates problems arising from the unequal expansion rates of materials with differing CTEs. While adhesive technology is still in its infancy when it comes to applications in the electronic assembly industry, by the year 2000 it promises to reduce the number of process steps, provide cheaper components and greater reliability, and decrease the environmental impact.

The challenges of supporting technologies

Thus far we have considered interconnection substrate technologies, known good die, and assembly technologies necessary to take us into the next generation of electronic products. We have focused especially on low-cost portable electronics. Several additional areas of packaging and interconnection technology, however, also demand significant attention and innovation.

Cooling of electronic components is becoming increasingly important. In many respects, our ability to effectively cool (or not effectively cool) limits the density to which we can place devices on a silicon chip. Cooling capability also determines how closely we can place those chips within a system. For a number of years, bipolar-silicon technology has been against the thermal wall, so to speak. Complementary metal-oxide semiconductor technology has now reached that same barrier.

We have developed impressive cooling systems for main-frame systems that can afford their cost both in dollars and inconvenience, but we often cannot extend these techniques to portable electronic systems. The amount of heat that can be conducted away from the chips and finally dumped into the air (or fuel in some cases) has also limited avionic systems. To simplify the cooling problem, we can divide it into



Figure 5. SEM of anisotropic conducting adhesive showing distribution of conductive particles.

two domains. The first involves spreading and pulling the heat away from the hot components (conduction). The second entails transferring that heat into some medium that will take the heat out of the electronic system (dissipation). Usually, the heat finally dissipates into the ambient air. The design of the system for adequate heat removal generally involves a careful interplay between the effectiveness of the conduction path compared to that of the final heat exchanger, or dissipation device.

The best ways of conducting heat from the chip involve an intimate mechanical connection to the chip. In many cases the mechanical connection to the chip for the heat path mechanically conflicts with the electrical connections to the chip; somewhere in the system there has to be a compliant member. Wire-bonding and face-up TAB technology are particularly attractive solutions to this problem. They employ compliant electrical connections in which the back of the chip can be conveniently clamped or glued to the heat-conducting substrate to achieve thermal impedances below 1 kW/cm^2 .

Unfortunately, such compliancy usually means "long and thin" electrical connections, which then has far-reaching consequences for the electrical fidelity of such connections. That is, high compliancy equals high electrical inductance measured in terms of nanohenrys, rather than a desirable few 10s of picohenrys. Flip-chip technologies are either limited in their heat conduction capability by the solder or adhesives, or they must have some compliant heat conduction mechanism connected to the back of the chip. The metal pistons used in the IBM thermal conduction module are such a mechanism.³ Many such mechanisms are impractical for portable, relatively high-G applications. Liquid conduction paths afford attractive opportunities by providing either direct immersion in the liquid or intimate, compliant conduction paths.

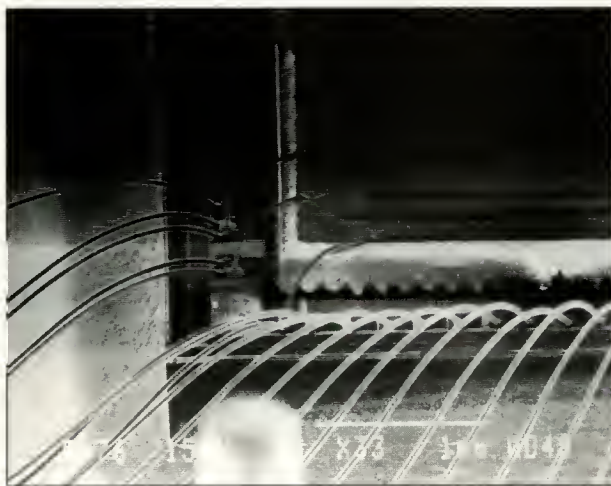


Figure 6. SEM of a close attachment capacitor.

The heat exchanger usually forms the weakest element in the process of taking the heat from the device and dumping it into the environment. Heroic efforts of forced convection can put air-cooling structures on a par with the conduction path in terms of overall thermal resistance from the device to the ambient (~ 1 W/K). But this last step usually has a thermal impedance orders of magnitude higher than the conduction impedance. For effectively cooling the chips, only simple convection can successfully spread the heat away from the chips into an ever-increasing surface area. We have only scratched the surface of ideas in this area, but heat exchange problems will likely limit the performance we can achieve from electronic systems for some time. Nature has much more to teach us in this regard.

Voltage disturbances caused by amps-per-nanosecond, high-current-demanding events on chips severely strain the power distribution, the regulation system, and the electrical "stiffness" required of the power connections to the chip. Lower power supply voltages, adopted to ameliorate both electric field reliability problems in submicron devices and the cooling problem, only exacerbate the power delivery problem. Techniques of making and exploiting local reservoirs of charge such as the MCC-developed technology¹ of close attachment capacitors (CACs) provides one way to manage the problem. We will soon need more exotic solutions, including active regulation on chip and more push-pull, off-chip switching events. Figure 6 shows a die with a CAC located on the active surface of the die. The thin-film capacitor is rated at only a few tens of nanofarads. We use conventional wire-bonding processes to attach it at multiple power and ground sites around the chip perimeter. Such techniques, exploited in SCP, can reduce power supply noise by an order of magnitude.

The development of the necessary software tools for the design of high-density, low-cost portable systems will call for substantial effort. As systems shrink in size, all aspects of the design and manufacture become very closely coupled. Electrical performance, power distribution, cooling performance, assembly methods, manufacturability, testability, and reliability all are interrelated. We need a good understanding of these interrelationships and effective tools to guide the generalist through all of the design options. Also the subject of intense work at MCC and its member companies, these developments exceed the scope of this introductory article.⁵

Technology development and transfer

I have outlined some of the technology developments necessary so that the North American electronics systems design and manufacturing industry can remain healthy. These developments are also necessary for this industry to regain some of the initiative it has lost concomitant with the earlier loss of the consumer electronics market. We are now poised to regain that initiative. This opportunity arises in part thanks to changes in the global economy that make it cost effective for North American companies to reenter the arena. Partly, too, we see a rapidly growing demand for a new set of consumer electronics products that will benefit substantially from North American innovation. Most significantly, we have all realized that we cannot do this alone, that cooperation is the way to win.

During the consortia efforts of the past decade, whether partly funded by government subsidies as with Sematech, or commercially funded efforts such as at MCC, we saw that such cooperation can accomplish important, commonly defined objectives in competitiveness. We made mistakes, and we learned from them. We have found that cooperation in basic technology research is significantly less costly and risky than individual company effort. We have discovered how to cooperate. Consortia soon could become a key way for electronics companies to conduct their research and development. Overall, consortia efforts to date would rate a B+ and are getting better.

The rationale for consortium research derives from a combination of technological and economic factors. Generally, this rationale includes such issues as

- promoting efficiencies of cost,
- sharing resources and risks,
- using special capabilities,
- pooling talent or facilities that cannot be easily supported by one company,
- diversifying a company's technology portfolio,
- providing a forum for the development of standards,
- developing a new technology through division of labor,
- organizing industry for faster technology development and greater competitiveness,

- developing infrastructure to support a new technology, and
- serving as a forum for technologists from different companies to gather and discuss similar technical interests that lead to strategic alliances.

The approximately 600 consortium research projects in Europe account for about 6 percent of Europe's R&D. Japan's 300 consortia account for around 4 percent of its total R&D. The US has only 60 or so, accounting for around 1 percent of R&D funding.

Organizing and running successful consortia is difficult work, comparable to setting up a new company. All the questions one must ask and answer in an ordinary business plan—delivery schedules, management compensation, market for use for the research, risks, competition—must be considered and addressed. In addition, many other issues are best resolved at the onset, including governance, funding, communication between members, technology transfer planning, staffing, and renewal of the research. Consortia such as MCC have successfully addressed these issues, and in so doing have provided a steady stream of technologies to the member companies.

In particular, MCC's Packaging/Interconnection Technology Program successfully developed and transferred to its 26 member companies technologies encompassing the entire packaging spectrum. MCC's nine years of research into, and development of, MCM technology has resulted in the fabrication of over 10,000 substrates, generally based on member-supplied designs. Figure 7 shows examples of high-density substrates and MCMs fabricated at MCC for both semiconducting and superconducting applications. We have developed substrates built upon metal, ceramic (Al_2O_3 and AlN), and silicon in sizes up to 6 inches². Many designs have exploited our proprietary QTAI design approach.

In addition, because of work on substrate and assembly technologies, members have incorporated such processes into production as the MCM substrate, TAB, wafer bumping, heat sink, CAC for noise reduction, and design software technologies. Also, thanks to this work, we have licensed such products to external companies as laser bonding and substrate test equipment. Supporting research in materials technology, design tools, test, thermal management, and key issues such as connectors have aided this work and greatly eased the transfer from technology research into member company products.

CURRENT WORK AT MCC FOCUSES on reducing costs for packaging and interconnections. As we have discussed, the driving technologies in packaging are now oriented around the requirement of portable electronics products. Conse-



Figure 7. High-density substrate and MCMs fabricated at MCC.

quently, MCC is pursuing work in the areas of low-cost, high-density, high-performance interconnection substrate technology, both infrastructure development for MCM-L and cost-reduction of MCM-D. In addition, projects specifically geared toward portable electronics establish technology baselines, forecasts, and design tools. They also investigate key technologies such as anisotropic conductive adhesives, flexible substrates, and high-density connectors.

Clearly, no single company can resolve the packaging and interconnection issues currently facing the industry. New technology design and supply infrastructure must be created. Cooperative research, development, and deployment of new technology is proving to be an effective way to rapidly and cost-effectively develop technologies needed for future projects and transfer them into the marketplace. As such, consortial efforts are becoming key enablers to our future technical leadership in this arena. ■

References

1. L. Higgins, "Perspectives on Multichip Modules: Substrate Alternatives," *Proc. IEEE Multichip Module Conf.*, CS Press, Los Alamitos, Calif., 1992, pp. 134-137.
2. D. Carey et al., "Variations in MCM Implementations Using Semi-custom MCM Technology," *Proc. Int'l Electronics Packaging Symp.*, 1991, pp. 94-106.

3. R. Tummala and E. Rymaszewski, *Microelectronics Packaging Handbook*, Van Nostrand Reinhold, New York, ISBN 0-442-20578-3, 1989, pp. 211-213.
4. H. Hashemi, P. Sandborn, D. Disko, and R. Evans, "The Close Attached Capacitor: A Solution to Switching Noise Problems," *Proc. 42nd Electronic Component Tech. Conf. (ECTC)*, May 1992, pp. 573-582.
5. P.A. Sandborn, "A Software Tool for Technology Trade-off Evaluation in Multichip Packaging," *Proc. 11th Int'l Electronics Mfg. Tech. Symp.*, pp. 337-341.



Dennis J. Herrell is vice president and executive director of MCC's High Value Electronics Division. He was a principal member of the technical staff in MCC's Packaging and Interconnect Program, responsible for the Multichip Systems Satellite. He spent 21 years at IBM where, most

recently, he worked at the T.J. Watson Research Center on Josephson technology and advanced silicon technologies.

Herrell has published over 40 technical articles and holds 15 patents. He received his PhD from Cambridge University and a BSc from Imperial College of Science and Technology. He is a senior member of the IEEE.

Direct questions about this article to Dennis Herrell at the Microelectronics and Computer Technology Corporation, 12100 Technology Road, Austin, Texas 78727-6298, or via e-mail at herrell@mcc.com.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 153

Medium 154

High 155

Search for Authors

IEEE Computer Society Press offers —

- * International promotion and distribution
- * Prestige
- * Competitive Royalties
- * A supportive development environment
- * High-quality production

We publish monographs, collections of readings, and tutorial texts for professional and educational markets as well as for our 100,000+ membership.



IEEE COMPUTER SOCIETY PRESS

10662 LOS VAQUEROS CIRCLE
LOS ALAMITOS, CA 90720

(714) 821-8380 • Fax: (714) 821-4010

CONTACT:
Henry Ayling, Editorial Director
h.ayling@compmail.com



THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.



Trends in Low-Cost, High-Performance Substrate Technology

Cost-performance pressures have increased demand for higher performance, higher density interconnection substrate technologies at little or no additional expense. The current shift to fine-pitch connections helps to drive the further development and cost reduction of multichip module technology. While ongoing increases in MCM-L laminate, PWB-type technology meet present needs, higher-density thin-film MCM-D (deposited) technologies may be required to keep pace in the future.

David H. Carey

Microelectronics and
Computer Technology
Corporation

High-performance packaging for multichip systems has received substantial attention recently because of the demands for lighter weight, reduced system size, higher system performance, improved reliability, and lower overall cost. Reports cite numerous multichip module strategies that use novel process technologies capable of achieving up to 1,000 inches of interconnection per square inch of interconnection substrate area (inch/inch²).^{1,2} By contrast, today's common printed circuit boards may provide only a small fraction of this wiring density. While the strict definition for an MCM has proven elusive, most MCMs are characterized by their use of fine-line wiring technologies and high active device to board area ratios.

Systems incorporating "bare die" chip mounting require high wiring densities. Eliminating the conventional single-chip packages of VLSI devices and attaching the die directly to interconnection substrates has greatly increased the I/O density per unit area. Use of "bare die" packaging can often lead to a tenfold higher I/O terminal density at the substrate level. A corresponding demand for increased interconnection density thus arises to achieve the necessary connectivity. Performance increases in integrated circuit technology will also demand the shorter, therefore faster,

interchip connections available through MCMs to realize the IC's full performance potential. Finally, using MCMs may improve reliability because this technology eliminates the added connections found in single-chip packaging approaches.

This article describes those trends in high density interconnection (HDI) MCM techniques that have the potential to reduce interconnection cost and production time. The implementation in MCM-L laminate technology of a workstation processor core will illustrate current substrate technology capabilities. All applications discussed are of a research nature, having been developed through the Microelectronics and Computer Technology Corporation's (MCC) High Value Electronics Division as ongoing R&D efforts supported by a number of North American companies.

Technology alternatives

Figure 1 shows a cost-density projection for the three primary MCM interconnection approaches. These technologies are defined as follows:

- ceramic multilayer interconnection (MCM-C), such as cofired multilayer ceramic (MLC),
- deposited dielectric (MCM-D), such as thin-film copper-polyimide (CuPI) and aluminum-SiO₂, and

- laminated dielectric (MCM-L), such as fineline printed circuit board.

Also included in Figure 1 is an approximate manufacturing cost objective we have set for these MCM interconnections. MCC anticipates reaching these cost objectives through MCM-L technologies (such as those described in this article), ad-

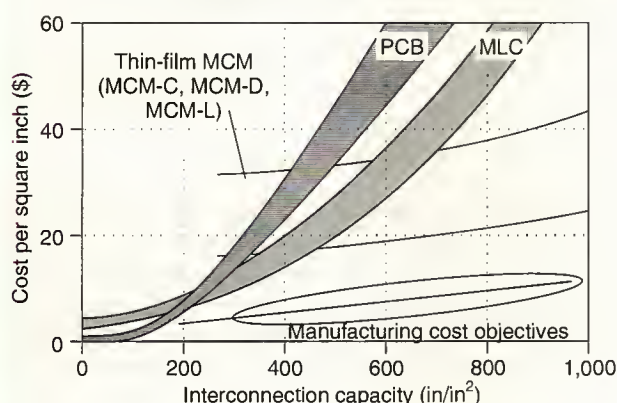


Figure 1. Cost-density curve for MLC, PCB, and thin-film interconnection substrates.

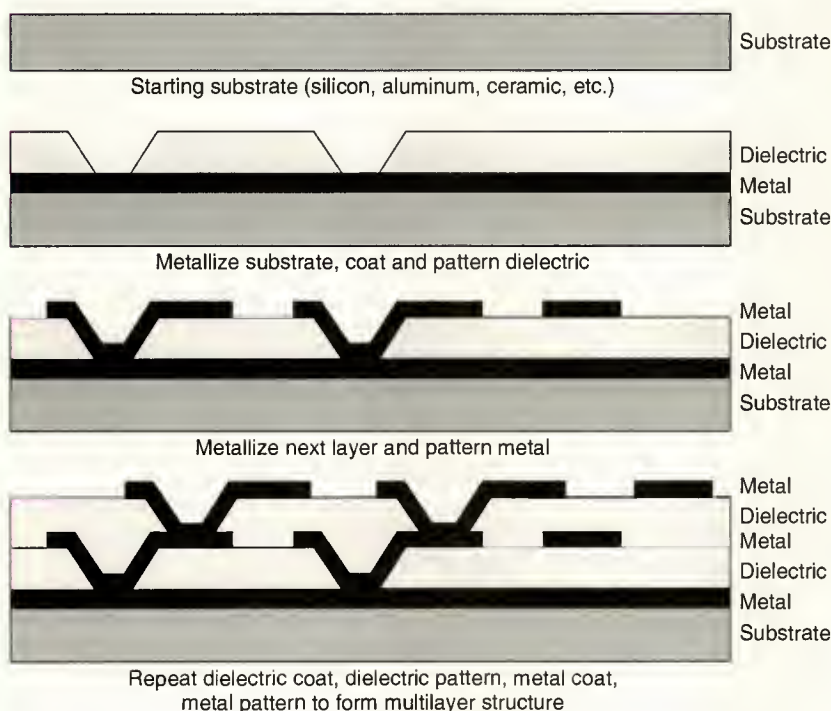


Figure 2. Generalized staggered conformal via MCM-D technology process.

vances in MCM-D process architectures, and possible combinations of these two technologies. MCM-C approaches may not offer as significant a potential for cost reduction as MCM-L and MCM-D technologies, and will not be considered here. Nevertheless, ceramics in combination with the thin-film and laminate technologies may provide an attractive cost-performance alternative (see "Hybrid solutions" later in the article). Our work has focused on creating a broad spectrum of solutions based on the MCM-L, MCM-D, and hybrid MCM-L/D solutions.

The thin-film technology MCM price bands are based on projected MCM-D interconnection costs, while the PCB and MLC bands represent current cost estimates. Due to the current low volume of MCM-D manufacturing and general cost-performance variations with PCB and MLC, these cost-density values should only be considered as a general comparison between the three technologies. Application-specific system factors dictate a thorough analysis to determine which technology represents the best value in a particular final product.³

We have been investigating low-cost, high-density MCM system technology in a number of research and development activities. MCC's thin-film MCM-D technology shows significant progress in reducing the cost of such interconnection structures. In addition, two projects have collaborated to investigate laminated MCM technology, or MCM-L, in high-

density systems, and so demonstrating cost-effective MCM-L technology implementation in a high-speed computer processor core.

Thin-film MCM-D technology

Much work has been done in recent years in developing MCM-D interconnection solutions that address future high-performance products. However, few of these technologies have yet proven to be cost-effective other than in applications in which the increased performance or reduced size offered by the MCM-D interconnection was beyond the scope of other approaches. Many current realizations of MCM-D technology have been directed toward high-end computer applications,^{4,5} although several companies are pursuing lower cost, high-volume applications.^{6,7} Many of these processes are based on aluminum or copper metallization that use sequential layers of insulators such as polyimide or silicon dioxide (SiO_2) and etched, conformal via technology. See Figure 2 for a generalized fabrication sequence for such a process.

A number of factors contribute to the relatively poor cost-effectiveness of current MCM-D technologies. Low total volume and low yield in the process are significant considerations, in addition to the lack of a vendor infrastructure to provide materials, equipment, and completed MCM-D substrates. Many MCM-D approaches are derived from semiconductor processes, yielding overkill capabilities for most applications. For example, many MCM-D processes can produce 10- μ m line widths, where 50 to 75 μ m (2 to 3 mils) is generally sufficient. The manufacturing processes thus utilize overly expensive capital equipment and require unnecessary clean room facilities. In addition, tested burned-in bare die are not readily available for use in the assembly of modules, which often significantly increases the overall system cost.

Indeed, several companies that entered the MCM-D business in the last few years have now left that arena, including APS/Raychem, Rogers Corporation, and Pacific Microelectronics Centre. The remaining companies, including nChip, MultiModule Corporation, CDI, and CTS, are working closely with industry, government, and consortia to hasten the technology's progress and acceptance.

Interconnection densities vary greatly among different applications and products. Many designs, such as for a processor and cache memory, are characterized by bimodal interconnection density requirements in which the complex chips demand very high density and high performance, yet are only a small portion of the design. A large fraction of the application (in terms of device count) needs far more modest routing density. Thus, the performance and density of thin-film MCM technology exceeds the "average" requirements for many applications. Consequently, we see a technology gap between the cost-to-performance ratios of low-cost, low-density PCB technology and high-cost, high-density MCM-D technology.

We must satisfy this broad range of interconnection densities required by varied applications, while also realizing a new, lower price-to-performance balance. Doing so will require that we explore a balanced effort in both higher density MCM-L and lower cost MCM-D technologies, in addition to mixed-technology solutions. Work in this area rapidly reveals that the economics of HDI fabrication is similar to that of both current PCB and IC production: large-format processing, with multiple parts per workpiece is vital to cost reduction. As will be illustrated later, MCC is currently investigating such techniques.

Extending PCB approaches

Given the high cost of MCM-D solutions, systems manufacturers are working closely with vendors and internal PCB shops to extend the limits of PCB processes into more sophisticated

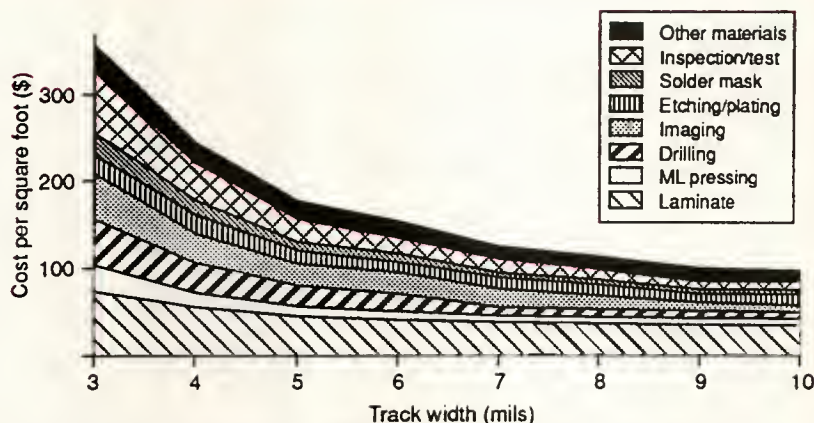


Figure 3. Process costs for 10-layer PCB. (Reprinted courtesy of BPA, Inc.)

MCM-L technologies. As a part of that cooperation, they are also addressing most commercial current and near-term high-density interconnection substrate requirements. A multitechnology approach has been necessary to demonstrate these improvements, as no single solution provides significant relief. Figure 3 illustrates the process cost factors of a typical PCB. While today's applications require track widths of 4, 3, or even 2 mils, Figure 3 suggests that substrate cost per unit area increases significantly as track widths decrease. Overall area reductions achieved by finer design rules may yet offset the cost increases.

The challenge of providing MCM-L technology with lines as fine as 2 mils at little or no cost increase requires attacking numerous areas in the fabrication of the substrate. There is no one "silver bullet" that will significantly reduce costs. Pains-taking work must proceed simultaneously in numerous process areas. The following key areas need simultaneous development for MCM-L applications:

- cheap, small vias (blind, buried through-holes for higher routing density),
- patterning and metallization technology for fine lines (volume production of 2 mil lines and spaces),
- thin precision laminates (for achieving scaled transmission line properties at finer lines and spaces),
- improved registration (to maintain large format processing with finer design rules), and
- better pad coplanarity.

MCC has explored low-cost, high-density MCM-L substrates through a test vehicle that provides the multichip integration of the central processor, math coprocessor unit, and cache static RAM of a high-performance reduced instruction-set computer workstation core. This workstation core example is typical of a current focus to integrate processor-cache clusters for ensuring the maximum performance possible with advances in IC tech-

nology. Previously, MCC fabricated several different-technology, thin-film MCM-D implementations of this workstation core and found that commercially available solutions were too expensive for use in the workstation market. This finding encouraged us to explore MCM-L solutions, which led to the implementation of the workstation processor core shown in Figure 4.

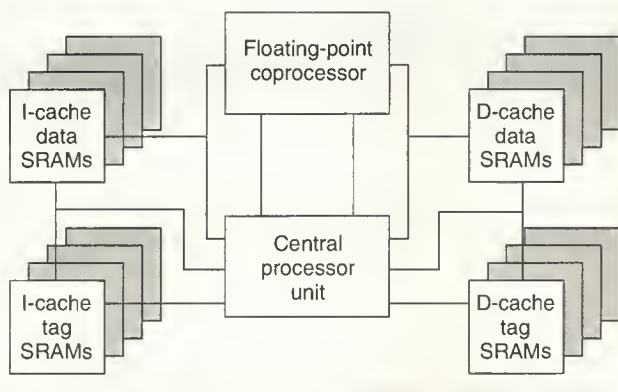


Figure 4. Block diagram of RISC workstation core used for test vehicle.

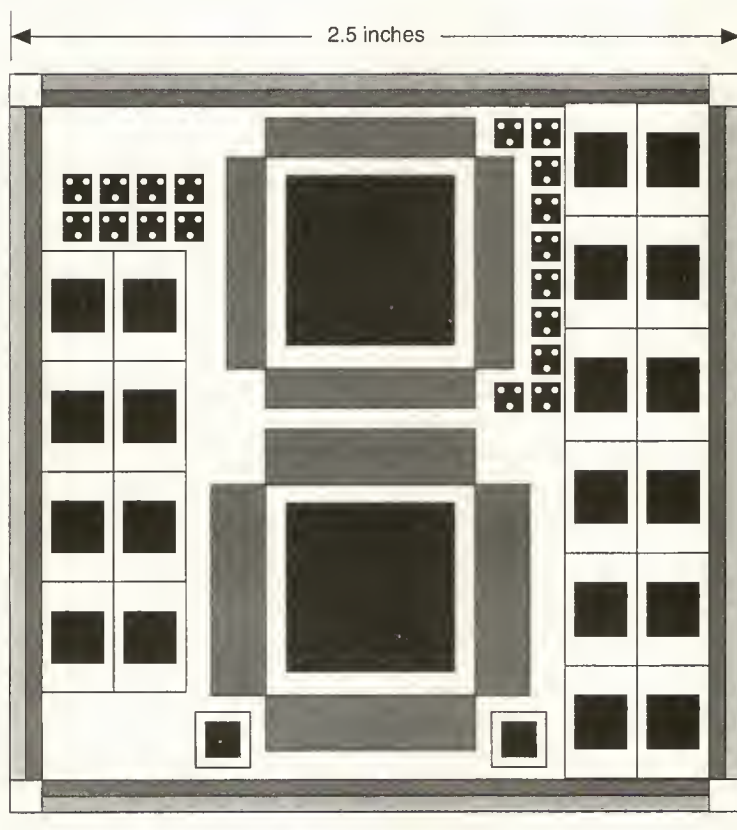


Figure 5. RISC processor core MCM-L component placement.

System specifications for the processor core chip set required a 50- to-100 MHz clock rate and subnanosecond edge rate performance from the interconnection. Anticipating no fundamental limitation on MCM-L interconnection performance based on these requirements, we executed a design predicated on the assumption of advanced PCB technology.

The complete processor core in this example contained the following components:

- CPU chip; $\sim 15 \text{ mm}^2$, over 400 I/O pins (~ 250 signal pads, 170 power/ground pads),
- FPU chip; $\sim 15 \text{ mm}^2$, over 250 I/O pins (~ 150 signal pads, 100 power/ground pads),
- 20 high-speed cache SRAM, over 30 pins,
- two bipolar clock buffer chips,
- discrete resistors,
- discrete capacitors,
- diode pack components,
- connector I/O sites, and
- test points.

Processor core design

Size constraints for the thin-film MCM-D implementation of the test vehicle and established connector footprints required that module size for the previously fabricated MCM-D implementations be set at 6.25 cm^2 (2.5 inch^2). All components had to be placed within $\sim 36 \text{ cm}^2$ (6 inch^2) of the interconnection. The design used a device placement based on the block diagram of Figure 4. See Figure 5 for a plan view of the approximate component placement.

For assembling the CPU devices, we selected a fine-pitch, $100\text{-}\mu\text{m}$ (4-mil) tape-automated bonding (TAB) technology that uses solder reflow. The FPU and SRAM had minimum lead pitches of $150 \text{ }\mu\text{m}$ (6 mil) and $300 \text{ }\mu\text{m}$ (12 mil). This assembly requirement dictated a $50\text{-}\mu\text{m}$ (2-mil) line-space interconnection capability to create the $100\text{-}\mu\text{m}$ (4-mil) pitch bond sites for the CPU. Routing studies and state-of-the-art PCB manufacturing capability led to additional design rule decisions for all other line widths and spaces. Final selected design rules were as follows:

- $50\text{-}\mu\text{m}$ (2-mil) line-space pairs for CPU bond pads,
- $150\text{-}\mu\text{m}$ (3-mil) minimum line-space pairs for all other outer and inner layer routes,
- $475\text{-}\mu\text{m}$ (19-mil) plated through-hole (PTH) pad size, and
- 1-mm (40-mil) PTH grid.

Summarized in Figure 6, the selected design rules for the test vehicle interconnection realized three tracks between vias. No buried vias were used.

Routing and layout

Device placement and design rule selection allowed for detailed routing. The complex and dense nature of the design required a combination of manual and automatic routing. We estimated that four routing layers would provide sufficient track density with the selected design rules before layout, although substantial hand routing was required to complete all connections. Further improvements in design and autorouting are needed to minimize layer counts, particularly given the more complex nature of MCM-L designs.

Four additional metal layers provide power distribution layers. These layers also serve as the transmission line reference planes for the signal traces, with laminate thicknesses selected to achieve the desired trace characteristic impedance. Figure 7 provides a cross-sectional diagram of the assignment chosen for each of the eight metal layers. Approximate laminate and prepreg (the glue or bond layers) thicknesses are shown. Note that the top and bottom layers are used for signal routing in conjunction with two internal planes, creating a mixed stripline and microstrip transmission line environment. Also note the use of back-to-back power-to-ground planes for higher distributed capacitance.

Several vendors fabricated the completed design, including both commercial PCB vendors and internal board shops of large systems houses. A high-temperature laminate material used on the board permitted a higher bonding temperature than with typical surface-mount technology (SMT). The design also specified a combination of plated solder and gold. The solder assembly process required that solder-plated pads be used for all TAB components. Gold-plated peripheral connector pads and wirebond sites for two buffer chips were required as well. The combination of fine lines and spaces and mixed metallization posed a number of fabrication challenges, although most vendors succeeded at least partially in building the design. Two vendors completed demonstration of the entire board; see Figure 8 (next page). Note that two parametric test chips replaced the coprocessor chip (optional in the system design) on the final fabricated board. Still, all design work was carried out for the more complex CPU/coprocessor design.

Thermal management

Figure 9 illustrates the two methods of thermal management integrated into the workstation core design. The design permitted both through-hole arrays and cutouts. A total module power dissipation of over 50 watts was required, with up to 15 watts generated by the VLSI devices. These power levels required a low thermal resistance through the board to a backside heat sink to limit IC junction temperature rise to

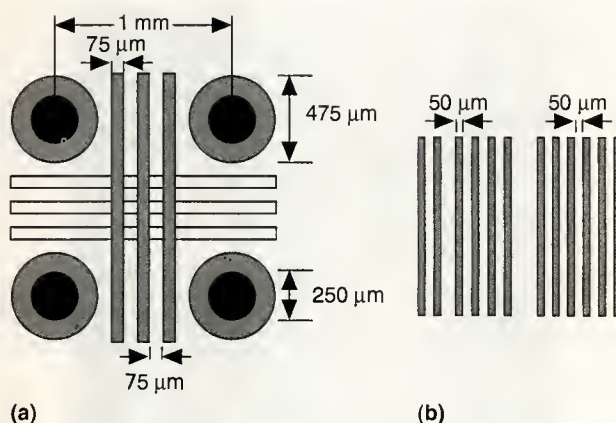


Figure 6. PCB design rules for MCM-L demonstrator: Primary design rules for all signal layers (a); design rules for 100- μ m OLB footprint and fan-out (b).

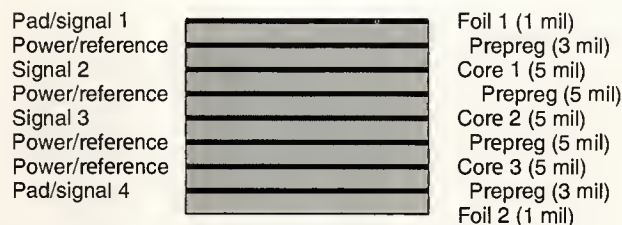


Figure 7. PCB cross section and layer assignment: board thickness of $\sim(5 \times 5) + (2 \times 3) + (2 \times 1) = 33$ mils.

55°C. We explored both thermal vias and thermal slugs in combination with a forced-air cooling heat exchanger as possible solutions.

We designed the thermal vias as 25-mil pitch through-holes, which precluded placing routing tracks in the region of the via arrays. This intentional aspect of the design allowed us to make a complete cutout under the die if the thermal via array resistance proved too high. We could insert a thermal slug that would make direct die backside contact through the cutouts and attach it to the cooling plate (assuming it was not an integral part of the heat sink itself). We found that high-density finned heat sinks and ducted airflow would be required to meet desired thermal impedances using the thermal vias. The thermal slug solution afforded us room for relaxation in fin density and airflow, although die attach and heat sink attach thermal resistances dominated the total thermal impedance from IC junction to ambient temperatures.

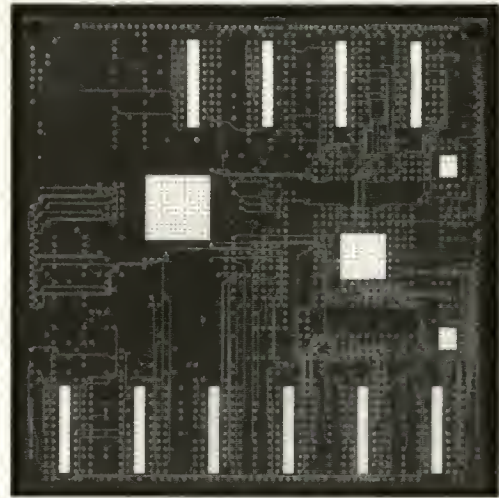
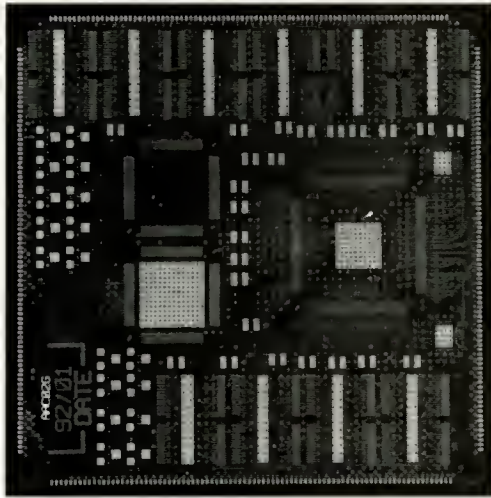


Figure 8. Completed workstation MCM-L PCBs.

The future of MCM-D

As system clock rates increase and routing-pin densities rise, MCM-L technology will invariably reach practical limits in some applications. In these circumstances, we may need to switch to MCM-D solutions. MCM-L may be challenged to cost-effectively extend beyond 500 inch/inch² wiring densities due to laminate thickness, line width, and via density limitations of conventional PCB technologies. Supporting high-performance clock rates (>200 MHz) and emerging direct chip attachment which meant techniques such as flip-chip (10-mil bond pad pitch) may also prove difficult with existing PCB methods. MCC's project work in PC workstation packaging suggests that performance requirements of such systems could typically be 150 MHz to 250 MHz by 1994, and greater than 300 MHz soon thereafter. Thin-film MCM-D substrate technology very likely will be an important part of addressing these forthcoming needs, but only if competitive costs can be realized. Similar to those required for MCM-L, we see a number of MCM-D technology developments needed to achieve lower cost:

- photo-imageable materials (for lower cost via and circuit pattern formation),
- low-cost metallization (to achieve thick, precise metal films),
- large-format processing (to achieve panelized MCM-D as discussed later),
- patterning, machining technologies (for ≤ 1 -mil lines, spaces, and vias over large area substrates), and
- process step and layer reduction, defect tolerance (to achieve the yield needed for low cost).

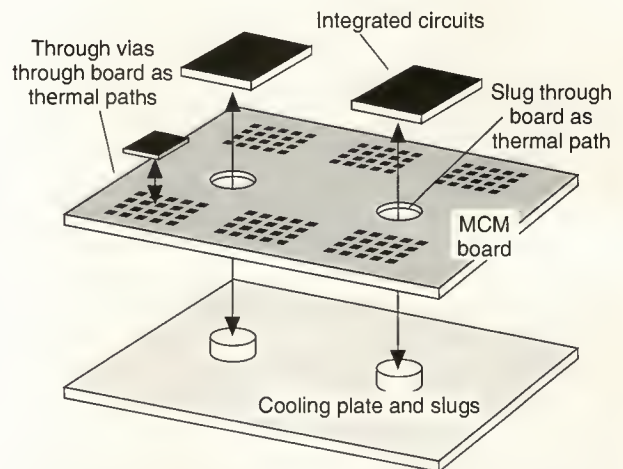


Figure 9. Thermal management of the MCM-L test vehicle.

Cost reduction of MCM-D technology requires that the lessons of both PCB and semiconductor fabrication be combined. Table 1 illustrates the characteristics of parts fabricated in both industries. In both cases, the workpiece-to-piecepart ratios (W/P) of 10:1 and 4:1, respectively, provide greater economies than the current 1:1 of most MCM-D manufacturing processes. In addition, the workpiece-to-feature (W/F) size ratios of 1,000s:1 also offer significantly greater economies than those of current thin-film implementations. Thus for MCM-D 25- μ m (1-mil) design rules and a typical 2-inch square

**Table 1. IC v. PCB industry
workpiece-to-piecepart, design rule ratios.**

Ratio	Integrated circuits	Printed circuit
W/P	150-mm wafer to 10- to 20-mm chip (10:1) (6-inch wafer, 400- to 800-mil chip)	600-mm panel to 150-mm board (4:1) (24x24-inch panel to 6-inch board)
W/F	150-mm wafer to 1.5- μ m features (100,000:1) (6-inch wafer, 0.06-mil features)	600-mm panel to 0.1-mm features (6000:1) (24x24-inch panel to 4-mil features)

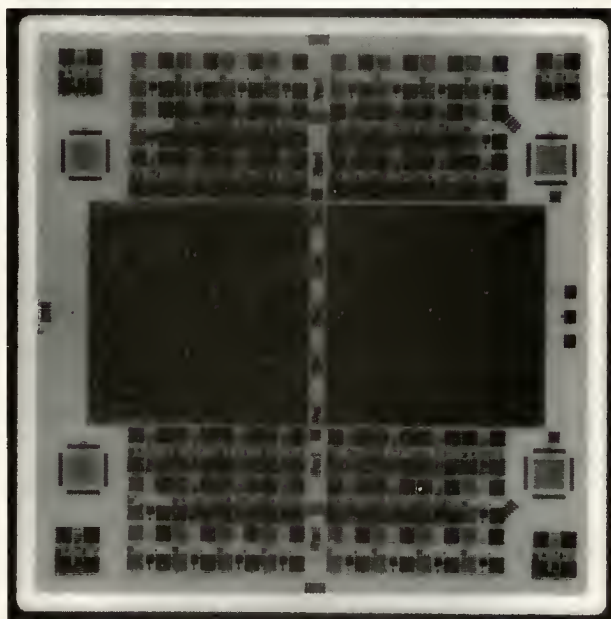


Figure 10. Six-inch square MCM-D substrate.

module size, a workpiece (panel) size of 6/100 inches provides equivalent ratios to these existing economies in PCB and IC fabrication.

MCC has adapted its thin-film MCM-D substrate manufacturing facility from 4-inch round wafers to 6-inch round and, recently, to 6-inch square wafers, demonstrating the feasibility of implementing multiple substrates on an individual workpiece. Note that the difference in active fabrication area

is ~6 to 7 inch² (4-inch round) versus ~30 inch² (6-inch square). Figure 10 shows a completed four-metal-layer, 6-inch square MCM-D process development test substrate with line widths and via sizes as small as 0.5 mils. Two separate 2.25-inch square MCMs and a number of test structures are incorporated into the test vehicle shown. In an actual product, the majority of the active area would be filled with multiple "copies" of functional MCM-D components.

MCC is continuing to develop a new thin-film process with approximately one half the process steps per layer of the older process, fewer clean room critical processes, and a larger base substrate format. We intend these processes to maintain capability for feature sizes (lines/spaces/via diameters) in the 10- μ m to 20- μ m range of past technologies. Completing this work will allow us to reduce the projected costs for this technology from those illustrated in Figure 11 (next page) for a five-layer MCM-D fabricated on a 4-inch diameter substrate to those illustrated in Figure 12 for a similar five-layer MCM on the 6-inch square workpiece (that is, "4-up" versus "1-up").

These cost estimates highlight the substantial savings made possible by larger base substrate sizes and improved process technologies. We are also exploring improvements that use layer count reduction techniques to provide full MCM-D functionality (routing, power distribution, controlled impedance) in as few as three patterned metal layers. Wiring densities however may be somewhat lower than the 1,000 inch/inch² anticipated in the five-layer design.

For additional cost savings, particularly in small quantities, MCC has developed a generic, semicustom interconnection substrate that can be customized to meet the needs of different chip sets.⁸ The primary element of the concept is a preproduced interconnection "blank" quick-turnaround interconnection (QTAI) that is later customized to produce a unique MCM, analogous to gate array and application-specific IC technologies in ICs. The use of an "interconnection array" provides opportunities for high-volume manufacturing of the generic component, thus reducing costs, improving yields, accelerating learning curves, and allowing process optimization. Also, because the customization sequence represents only a small fraction of the overall interconnection fabrication, we may achieve rapid part turnaround versus a fully custom design.

Hybrid solutions

Interesting possibilities exist for hybrid approaches that combine MCM-C, MCM-D, and MCM-L manufacturing approaches. For example, sequentially manufactured, fine-line microvia structures (~2-mil to 4-mil lines/spaces/via diameters) built up on PCB bases may offer key attributes of both technologies. Such an MCM-D/L approach may optimize cost-performance trade-offs by providing improved via densities and PCB processing economies. There are a number of current examples of MCM-D and MCM-C being combined to

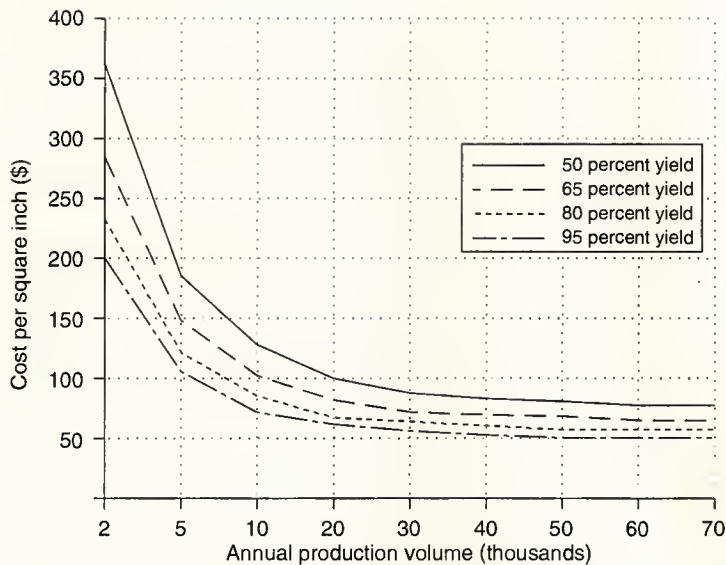


Figure 11. Cost estimate for five-layer MCM on 4-inch round substrate using MCC baseline MCM-D process: 4-inch diameter alumina substrate, five metal layers (two power/ground, two signal, one bond pad), copper conductors, polyimide dielectric, 5-inch square usable area.

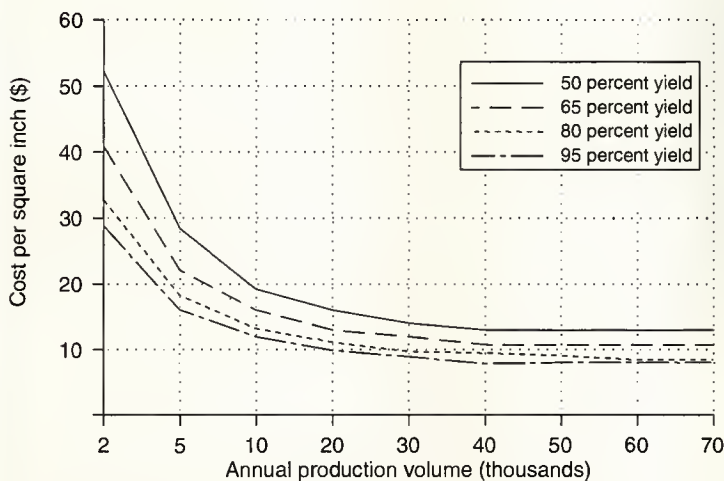


Figure 12. Cost estimate for five-layer MCM on 6x6-inch square substrate using improved MCM-D process: 6x6-inch alumina substrates, five metal layers (two power/ground, two signal, one bond pad) copper conductors, polyimide dielectric, 30-inch square usable area.

leverage the advantages of each technology. Such hybrid combinations may provide the best strategy for reducing cost, assuming manufacturing challenges of integrating differing technologies can be overcome.

WE MUST PROVIDE COST-EFFECTIVE, high-density interconnections within a range of options, each addressing a different cost-performance trade-off in a given design. The evolution of PCB technology into more sophisticated MCM-L offers a promising technology to address not only low-end capabilities but also high-density and high-performance requirements. For some applications, improved thin-film MCM-D technology can ultimately be a cost-performance winner, once we resolve current manufacturing cost issues and achieve high volume production. In addition, current research is pursuing techniques that marry classical thin-film and PCB approaches, two typically isolated manufacturing technologies. The careful combination of these methods may strike the best cost-performance balance.

MCC is making ongoing efforts to build on MCM-L and MCM-D demonstration vehicles and to further push the cost-performance capability of both techniques. Work also progresses on new MCM-D technologies that could be cost competitive with MCM-L and MCM-C approaches. While the supporting infrastructure for MCM-D is still weak, we anticipate that an application space for these more advanced technologies will emerge before the end of the decade. To address the broad spectrum of application requirements and cost-performance targets, we believe that both MCM-D and MCM-L technologies (perhaps in careful combination with ceramics) will be required. ■

References

1. J. Hagge, "Ultra-Reliable HWSI with Aluminum Nitride Packaging," *Proc. Nepcon West*, March 6-9, 1989, pp. 1271-1283.
2. C. Ho, D. Chance, C. Bajorek, and R. Acosta, "The Thin-Film Module as a High-Performance Semiconductor Package," *IBM J. Res. Develop.*, Vol. 26, No. 3, May 1982, pp. 286-296.
3. J. Shiao and D. Nguyen, "Performance Modeling of a Cache System with Three Interconnection Technologies: Cyanate Ester PCB, Chip-on-Board, and Cu/PI MCM," *Proc. IEEE MCM Conf. Proc.*, pp. 134-137.
4. R. Tummala, H. Potts, and S. Ahmed, "Packaging Technology for IBM's Latest Mainframe Computers

(S/390/ES9000)," *Proc. 41st Electronic Component Technology Conf.*, 1991, p. 682.

5. D. Akihiro, W. Toshihiko, and N. Hideki, "Packaging Technology for the NEX SX-3/SX-X Supercomputer," *Proc. EIA/IEEE Electronic Component Technology Conf.*, 1990, p. 525.
6. U. Deshpande, S. Shamouilian, and G. Howell, "High Density Interconnect for Vax 9000 System," *Proc. Int'l Electronics Packaging Society (IEPS) Conf.*, 1990, p. 46.
7. B. McWilliams, "Comparison of Multichip Interconnect Technologies," *IEPS Proc.*, Sept. 15-18, 1991.
8. D. Carey et al., "Variations in MCM Implementations Using a Semi-Custom MCM Technology," *Proc. IEPS Conf.*, Sept. 15-18, 1991, pp. 94-106.



David H. Carey is a senior member of the technical staff with the MCC Packaging/Interconnection (HVE) Program. His interests include low-noise power distribution for high-performance computer packages, high-frequency analysis, package concepts for multichip systems, and

MCM design and processes. Currently, he is group leader of the Microfabrication Process R&D group.

Carey earned the BSEE and MSEE from Texas A&M University and is a member of Tau Beta Pi, Eta Kappa Nu, IEEE-Components, Hybrids, and Manufacturing Technology, and IEPS.

Direct questions about this article to David Carey at the Microelectronics and Computer Technology Corporation, 12100 Technology Road, Austin, TX 78727-6298; or via e-mail at misunas@mcc.com.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate numbers on the Reader Service Card

Low 156

Medium 157

High 158

NEW TITLES from IEEE Computer Society Press

CURRENT RESEARCH IN DECISION SUPPORT TECHNOLOGY

edited by Robert W. Blanning and David R. King

This new book contains an introduction and fifteen selected papers revised for publication. Its articles are divided into three sections each corresponding to one of the three major areas of current research.

The first area is advanced decision modeling and model management and covers recent research on logic modeling, model integration, the economics of designing and using systems containing models and other DSS tools. The next section explores the use of expert systems in DSS construction, the application of connectionist architectures, and the development of active DSS that adapt to the needs of their users. The last section discusses group DSS, the determination of the organizational impact of DSS, and the application of computer and cognitive science concepts to understanding of organizational information processing and decision-making.

Sections: Introduction, Advanced Decision Models and Model Management, Knowledge-Based Decision Support, Behavioral Issues in DSS Development.

256 pages. March 1993. Hardcover. ISBN 0-8186-2807-3.
Catalog # 2807-01 — \$45.00 Members \$35.00

SOFTWARE REENGINEERING

edited by Robert S. Arnold

The tutorial introduces software reengineering definitions, themes, technology, strategies, and risks, and describes software reengineering concepts and processes, tools and techniques, capabilities and limitations, risks and benefits, research possibilities, and case studies

Various key sections of the text present, evaluate, and examine:

- * Several examples of real-life reengineering projects
- * Reengineering and CASE tools
- * View-based systems, modularization, and transformations
- * Data reengineering and approaches for migrating data
- * Tools and approaches for decomposing programs
- * Processes, metrics, and source codes
- * Knowledge-bases and architectures for software reengineering and reverse engineering

Sections: Business Process Reengineering, Strategies and Economics, Reengineering Experience, Technology for Reengineering, Data Reengineering and Migration, Source Code Analysis, Software Restructuring and Translation, Reengineering for Reuse, Reverse Engineering and Design Recovery, Object Recovery, Knowledge-Based Program Analysis.

c.600 pages. November 1992. Hardcover. ISBN 0-8186-3272-0.
Catalog # 3272-01 — * \$75.00 Members \$60.00
(* prepublication price)



To order call toll-free 1-800-CS-BOOKS

or 714/ 821-8380 or FAX 714/ 821-4010





Programmable Multichip Modules

The MCM approach to electronic packaging improves system performance by bridging the gap between the current packaging technology and advancing IC technology. However, without a mature infrastructure, a fully customized MCM design takes significant engineering effort and incurs costly fabrication fees and cost increases for each mask layer. Programmable MCMs—wafers with sites for chips and several layers of programmable interconnections—minimize the engineering delays and the cost.

Surendra Burman

Vista Technologies, Inc.

Naveed A. Sherwani

*Western Michigan
University*

Multichip modules represent a new approach to interconnecting chips that replaces conventional printed circuit board technology in high-performance applications. Bare chips are interconnected on a substrate with substantially finer conductor lines, smaller dielectric thickness, and denser via grid than on a PCB. As a result, MCMs are not subject to the conventional PCB design rules and assembly restrictions. Figure 1 shows a typical MCM.

As indicated in Figure 1, metal wires laid out on layers called signal planes connect the chips. Different layers make up the horizontal and vertical connections, and additional layers provide power and ground connections. Several dozen layers may be used as signal planes.

The MCM approach to the current packaging technology speeds electronic systems by translating semiconductor speed into system performance. However, due to the lack of a mature infrastructure, high-density and high-performance MCMs are expensive to fabricate. The cost of fabricating an MCM is directly proportional to the number of mask layers.

In some applications the engineering delays in designing and fabricating such modules become unacceptable. This need for quick turnaround time plus volume production, high product yield, and

low cost has led to the development of another approach called programmable multichip modules (PMCMs).¹ In the PMCM approach, a fabricated generic substrate is customized to meet the application-specific needs of a user. PMCMs permit fast turnaround and make the modules cost-competitive with conventional packaging.^{2,3} Recently, this semicustom approach to MCMs has drawn the attention of MCM manufacturers. While PMCM technology reduces the cost of MCM development by a significant margin, it also degrades system performance in comparison with fully customized MCMs, due to the electrical effects of the programmable elements.

The main difference between a PMCM and an MCM lies in the nature of the interconnection. A PMCM comes with prefabricated interconnection layers that let us customize the layers based on the user's needs. Depending on the customization process, we can classify the PMCMs into two types, fully programmable and semiprogrammable.

The customization process in fully PMCMs can involve either electrically programmable or laser-programmable technologies. They require no user-defined fabrication. The fully programmable MCM (FP-MCM) approach is similar to the field programmable gate array (FPGA) approach.⁴ The semiprogrammable approach conceptually compares to the conventional gate array approach. A

semiprogrammable MCM (SP-MCM) requires user-defined fabrication only in the last few layers.

A typical FP-MCM is a silicon wafer that provides I/O buffer sites for chips, electrically programmable connectivity between these buffer sites, and a base for other components. Bare chips are mounted and wirebonded to the wafer; setting electrically programmable switches interconnects the chips. A switch is essentially a programmable via since it connects a horizontal metal line to a vertical metal line.

A more recent development in FP-MCM technology uses a laser-programmable technique to make connections.⁵ A laser-programmable MCM consists of a silicon substrate with a dense, predefined array of pads, tracks, and links. IC chips are mounted, circuit side up, on the substrate, and the chip pads are wire-bonded to the substrate pads. Links can be formed at many of the interconnections of vertical tracks with horizontal tracks. A laser pulse applied to a link region forms a conductive vertical path (or via). A laser can also cut tracks to isolate distinct nets or trim unwanted stubs from a net. Although FP-MCM technology allows fast turnaround, it usually comes at the cost of the performance.

An SP-MCM design consists of a prefabricated, blank-interconnection array in a copper polyimide (CuPI) substrate. Several routed layers, only a few of which are fabricated based on a user's specification, make up the substrate. The existing substrate layers can be programmed based on a description of the chip pad placement and a netlist. This type of architecture, called quick-turnaround interconnection (QTAI), reduces turnaround time by approximately 60 percent to 70 percent.

The ends of all the segments in the prefabricated layers are brought to the top and can be interconnected using additional layers. To achieve design-specific interconnection, we fabricate two layers of short segments on top of the prefabricated layers. Since this approach requires customization of only the last few layers (1-3), it is considered to be a semiprogrammable one. SP-MCMs avoid the performance penalty of an FP-MCM, as they do not use programmable switches. However, it does take more time to implement a design using an SP-MCM when compared to an FP-MCM.

PMCM design principles

Since a complete custom design of an MCM is very expensive and usually requires many weeks of engineering effort, a need exists for an MCM generic substrate with prefabricated interconnections. This approach allows the user to place chips and make connections between them.

The PMCM approach is rapidly gaining ground because of its fast turnaround time, high yield, and low cost. It is similar to the FPGA technique of prefabricating all or most of the circuit and later customizing the device to meet the application-specific needs of the user. Essentially, a PMCM contains a prefabricated blank substrate with programmable interconnections. The wiring segments are fabricated with electrically

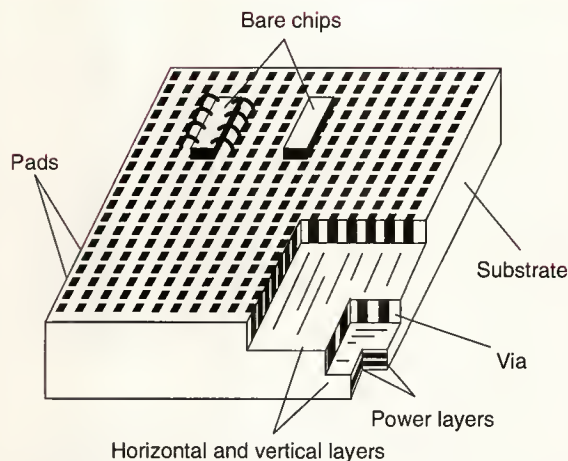


Figure 1. A multichip module.

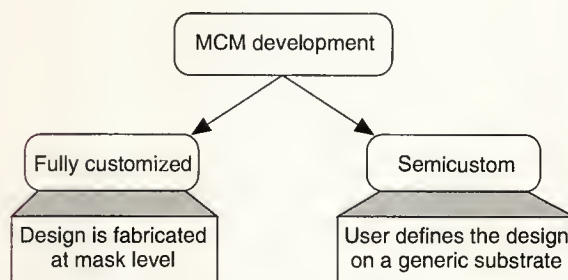


Figure 2. Fully customized versus semicustom MCMs.

programmable or laser-programmable switches. This technique results in the manufacture of a generic substrate that can be mass produced with high yield. Later, the substrate can be customized by making connections with the help of suitable voltages applied across the programmable switches or laser. Figure 2 illustrates the underlying principles of semicustom and fully customized approaches to MCMs.

The fast product turnaround time of PMCMs results from two facts. A substantial fraction (if not all) of the substrate and interconnection is prefabricated, and much of the design, tooling, and fabrication time is avoided for each individual application. The programmable approach can provide a predictable and well-characterized MCM technology. PMCM technology provides the user with a presimulated or precharacterized substrate and interconnection that can be of enormous value in the design and simulation of the final product. Substantial effort is usually required in a fully customized design environment to characterize and simulate the expected behavior of the MCM. This behavior ultimately trans-

lates into higher cost and additional design time. Because of the PMCMs' ability to provide manufacturing leverage (lower cost) and rapid product realization, designers need to consider the technical issues, mainly, two important aspects: routability, and electrical design and performance.⁶

Routability. Like gate arrays, routability is a key factor in the design of PMCMs. Designers must define most, if not all, of the masking or phototooling steps in the beginning, before designing the system. A substrate is manufactured in a generic fashion and then customized to meet the specific needs of the user. The ability to route complex and dense multichip designs imposes the requirements for an early design of a highly routable wiring topology.

Basically, we achieve good routability in two ways. We can use a highly efficient layout of the base substrate and high wire accessibility to allow great flexibility in configuring wire resources. Or we can use modest routability and wire access combined with an abundance of wiring made possible by the lithographic techniques used in most MCM manufacturing processes. These two techniques imply that the method chosen to achieve good routability involves a series of trade-offs. For example, if we choose to fabricate more wiring to overcome limitations in routability, we push the design rules and manufacturability.

An important component in achieving efficient programmable designs is the design tool that can decipher the programmable wiring structure and perform the actual customization (routing) needed to realize an application-specific MCM. Note that the routing efficiency is a factor of both the base wiring density (typically measured in inches of wire per square inch of the substrate area) and the resource utilization (that is, what fraction of the available wiring can be used in routing a design). We must account for the total wire length used, relative to the minimum theoretical routing length. This difference describes any deviation produced by the routing and avoids including any wire length associated with "stubs," which are essentially wasted resources even though they may be used in the routing.

Electrical design and performance. Another important requirement for PMCM design is electrical performance. If suitable performance cannot be realized with the programmable approach, we will not be able to meet the application objectives. However, it is not easy to define the suitable measure of performance. The wide array of circuit designs that users may want to map onto an MCM provides the supplier of the PMCM with a dilemma. The question as to how much performance can be guaranteed is a difficult one. Equally difficult is the question as to what level of performance will satisfy the various market segments.

In many circumstances, electrical performance of the signal interconnection will be relatively good even without rigorous design for characteristic impedance, low loss, stub minimization, and so on. Performance is good because the

electrical length of the signal wiring in most MCM environments is short compared to the wavelength/rise times of the IC signals.

The main issue in many cases is capacitive loading reduction for CMOS systems to minimize delay caused by RC time constraints. In other words, a large fraction of system designs will need to address signal delay more than high-bandwidth signal fidelity. This may not be the case in a more conventional, single-chip packaging/PC board implementation in which physical/electrical lengths of interconnections are longer and more significant. Perhaps the more pressing issue related to signal fidelity is power distribution. Many signal noise problems are related to the absence of clean power and ground supplies from which noise is fed forward through output drivers, and noise margins are diminished at the receivers. These concerns place an additional demand on the design of a PMCM. Not only must the power distribution scheme be flexible, it must also support high performance. The usually predefined power distribution network of the MCM design accommodates a variety of supply voltages, varying numbers of supply potentials, and a variety of both AC/DC current requirements.

In PMCM design, the power distribution is often predefined along with the signal wiring network. As an option to the designer one could define a family of part types to address differing electrical performance requirements for both signal and power distribution while still building on the same basic design concept. The obvious danger of such a strategy lies in the proliferation of part types that begins to diminish the advantage of the PMCM approach.

Fully programmable MCMs

As discussed earlier, an FP-MCM contains a programmable substrate. We classify FP-MCMs on the basis of the programming method used to customize the interconnection: electrical or laser.

Electrical programming. The FP-MCM architecture developed at ERIM^{7,9} contains 4-inch silicon wafers called programmable silicon circuit boards (PSCBs). They mainly provide

- I/O buffer sites for chips to interface externally to the rest of the system,
- electrically programmable connectivity between these buffer sites and the generalized die attachment area on the wafer, and
- a base for which all components can be attached.

The FP-MCM developed at ERIM supports four "daughter" pieces of silicon called segments, which provide bonding sites for the various ICs. These segments, one per quadrant, are attached to the base wafer, and are surrounded by the I/O buffer sites and the interconnection area (See Figure 3). This interconnection area, or trench, supports intersegment

Programmable interconnections

Four metal layers separated by dielectric layers make up the substrates. Power distribution takes place in the two lower layers, while the upper two layers supply an orthogonal wiring grid with permanent vias or antifuses in selected grid interconnections. The top layer also includes the bonding pads to which the bare chips are electrically connected after programming completes.

A signal path can be programmed through the substrate by linking previously uncommitted line elements together by the antifuses. The interconnection line architecture of actual designs is much more complicated, but the principle of programming is the same in either case. Because all line elements are in one way or the other accessible from a bonding pad, we can use a wafer prober to apply a programming pulse with a voltage amplitude larger than the threshold voltage to a pair of wiring elements to connect them to each other. The programming pulse is composed of a positive and a negative half pulse of such magnitudes that the total pulse is larger, but a half pulse is smaller than the threshold pulse.

Since all wiring elements, except the two which are to be linked together, are kept at ground potential, antifuses that are not intended to be fired are at worst exposed to half the threshold voltage and therefore safe from accidental firing. Theoretically, the connections from bonding pads to either the voltage or the ground plane could also be made programmable. However other design constraints have so far stood in the way of power programmability. Instead, a large number of pads with permanent vias to voltage or ground are spread over the substrate in such a way that the power pad can always be found very close to where it is needed.

References

1. H. Stopper, L. Harris, and S. Choksi, "High Density Multichip Modules Using an Electrically Programmable Interconnection Substrate," *Proc. NEPCON East*, 1991, pp. 189-198.

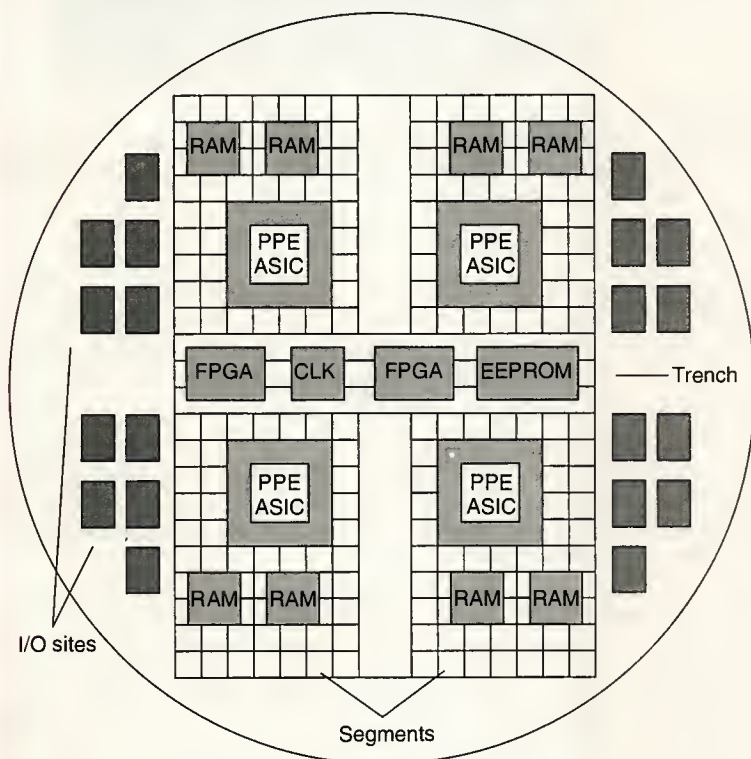


Figure 3. A fully programmable MCM.

connections and limited chip bonding sites. Combining four segments with a base wafer leads to increased yield of the assembled wafer as compared to the yield possible when all structures are placed on a monolithic wafer.

The key feature of ERIM's PMCM process is that the interconnections on the substrates and the segments are programmable. The MCM process employs wire bonding for connecting chips to the programmable substrate. The interconnections (see the Programmable Interconnection box) in the substrate are laid out in a two metal layers of rows and columns across the wafer and are electrically connected at intersections via a programmable antifuse.⁸ (See Antifuse box, next page.) An electrical programming process that converts high-resistance antifuses to low-resistance conductive paths adds the design-specific information after wafer fabrication.

ERIM uses these PMCMs to implement ASIC designs. Recently, designers demonstrated this technology on a 100,000-gate image memory controller and a 120,000-gate pipeline processing element. Banker discusses the various electrical, test, speed, and thermal design issues considered for implementing the ASIC designs.⁷

Antifuse

Initially the antifuse is a very large resistor (off state). Application of a predetermined threshold voltage changes it into a small resistor (on state). Because of a permanent structural change, once the switch is programmed, it cannot be switched back into its original state.

The literature describes several types of antifuses for various applications.¹⁻⁵ The type used in the MCM process is composed of a sandwich of metal, amorphous silicon, and metal, as shown in Figure A. The metal layers are not different from those used to incorporate the wiring. Note however that the antifuse does not occupy more area than a permanent via and that both are confined within the normal line width of the conductors to which they are attached.

The amorphous silicon has very high resistivity and thus is practically an insulator. This fact explains the initial off state of the antifuse. The on-state resistance attained with a first fusing pulse of a certain amplitude may be reduced further by a second fusing pulse of larger amplitude, but it cannot be reduced by a subsequent pulse with a smaller amplitude.

2. E. Hamdy et al., "Dielectric-Based Antifuse for Logic and Memory ICs," *Proc. Tech. Digest-Int'l Electron Devices Meeting*, 1988, pp. 786-789.
3. H. Stopper, "A Wafer with Electrically Programmable Interconnections," *Digest Tech. Papers, 1985 Int'l Solid-State Circuits*, 1985, pp. 268-269.
4. M. Lunnan and D. Greve, "The Microstructure of Programmed n+ p n+ Polycrystalline Silicon Antifuses," *J. Applied Physics*, Vol. 54, No. 6, 1983, pp. 3278-3281.
5. R. Cooley, "Control Video Timing with an Antifuse Array," *Electronic Design*, Vol. 37, No. 12, June 1989, pp. 59-62.

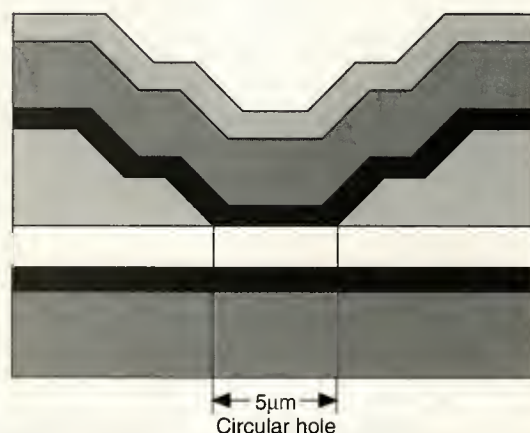


Figure A. Cutaway model of the antifuse.

References

1. K. El-Ayat et al., "A CMOS Electrically Configurable Gate Array," *Digest Tech. Papers, 1988 Int'l Solid-State Circuits Conf.*, Vol. 309, 1988, pp. 76-77.

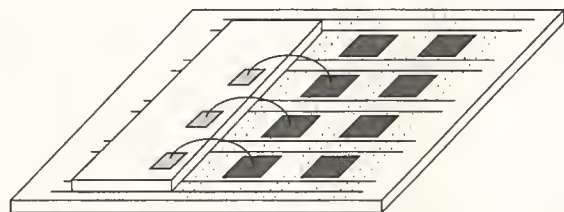


Figure 4. A laser-programmable MCM.

Laser programming. Berger discusses a laser PMCM under development at MIT's Lincoln Laboratory.⁵ It is a silicon substrate with a dense, predefined array of pads, tracks, and links, as shown in Figure 4. The substrates are manufactured in advance and then quickly programmed by laser to implement a particular system. The 10-cm wafer module contains pads and tracks over an area 5-cm square, two layers of metal,

and no active circuits. Power, ground, and signals share the tracks on each metal layer. The module holds an array of 258-by-258 pads on a 200- μ m pitch and five tracks between two adjacent pads. In Figure 4, the first, third, and fifth tracks between two pads support signals; the second and fourth support power and ground. The latter are preconnected into power grids with direct, metal-to-metal vias (small squares with diamonds). Although the module has no solid ground planes, every signal path is adjacent throughout its length to a power track.

ICs are mounted and wire-bonded to the substrate pads. Horizontal tracks are on one level of metal and vertical tracks on the other. Links can be formed at many of the intersections of vertical tracks with horizontal tracks. These links consist of a silicon nitride layer sandwiched between the two metal layers. Details of the layout are shown in Figure 4. Directing a laser pulse, of the correct power and duration to the link region, programs a link. The metal and the nitride fuse to form a conductive vertical path. The laser also cuts

tracks to isolate distinct nets and trim unwanted stubs from a net. Cut points are the narrow places in the tracks.

Thicker metal gives lower resistance, and a thicker insulator produces lower capacitance. Both improve the speed of signal propagation across the module. The thickness for intermetal insulation is 1.5 μm of nitride at the link areas, augmented by 2.0 μm of oxide elsewhere. The target thickness for metal is 2.0 μm . These values would give a resistance of about 140 ohms and a capacitance of about 15 pf for a 5-cm track 16 μm wide. Such a track should allow propagation across the whole wafer in about 1.1 ns, which should support signals switching at 100 MHz.

Before MCM substrates can be used, they must be tested for continuity of the tracks and freedom from shorts between crossing tracks. To allow probing for these tests, we connect a pad to each end of each of the 2,580 tracks by replacing some of the links along the edges and in the corners of the MCM with vias. Testing on a wafer prober involves measuring the capacitance between the probed track and the silicon substrate. A capacitance lower than normal indicates a break in the track; one higher than normal indicates a short to some other structure. If a signal track is found to be open or shorted to an adjacent (power) track, it cannot be used for routing signals. Some other track can probably be used. However, if the two power grids are shorted together, the whole MCM is unusable.

Most of the existing tools developed for wafer scale integration are being used for laser PMCMs.⁹ They facilitate 1) placing chips on the substrate, 2) routing of the interconnection between the chips, 3) controlling the laser system in the linking and cutting operations, and 4) generating test patterns.

Although FP-MCMs provide fast turnaround time and low cost, they usually affect performance due to the poor electrical behavior and programmable nature of the antifuse. Thus this technology is limited to implementation of low- to medium-performance systems:

Semiprogrammable MCMs

Performance degradation is the major drawback of FP-MCMs. MCC develops the semiprogrammable MCMs called QTAIs as a compromise between FP-MCMs and fully customized MCMs.⁶ QTAI technology reduces costs and offers faster turnaround times than fully customized MCMs and does not severely degrade performance over fully customized MCMs. This technology uses a prefabricated CuPI substrate with a blank-interconnection array. The customization process, which involves only the fabrication of very few interconnection layers and possibly a bond pad layer, forms the final product. The generic substrates can be mass produced, and customization involves only a fraction of the total fabrication of a custom MCM.

Since most parts of the system are generic, the fabricating process can be highly optimized, thus increasing the yield and reducing the cost. Avoiding much of the design, fabrica-

The major difference between fully customized MCMs and QTAIs is interconnection routing.

tion, and testing for each application reduces the turnaround time by 60 percent to 70 percent. When compared to the FP-MCMs, QTAIs have greater flexibility in the customization process.

The FP-MCM approach is similar to the FPGA approach, as is the customization process. Customization is carried out by setting electrically programmable switches to establish the connectivity needed by the user. On the other hand, the QTAI approach is similar to the gate array design style. More accurately, it is like an "interconnection array." An array of short wire segments with open ends are prefabricated, and the interconnection array then is customized to form the final product. The customization involves fabricating the interconnection layer and the bond pad layer.

The major difference between fully customized MCMs and QTAIs is the routing of interconnections. While all the interconnections in fully customized MCMs need to be customized on the whole 3D routing space, the routing in QTAIs takes place in two stages. During the first stage, many short wire segments are prefabricated, a generic step for all applications that must be carefully designed. For example, the conductor layer and conductor layer 2 in Figure 5 are prefabricated and used for x-direction wire segments and y-direction wire segments, whose ends are open and are brought to the terminals on a third layer.

During customization, connecting the terminals on the third layer using short conductor links completes the routing of the netlist of MCM design. Users specify the interconnection pattern on the third layer to serve their application's need. Figure 5 compares the manufacturing sequence for fully customized MCMs and QTAI technology.

A key issue in designing the interconnection for QTAI is the choice of a small number of part types that can be produced in high volume to serve a large number of different applications. Good routability serves all kinds of applications. Trade-offs among various factors have to be made. For example, prefabricating more wiring to improve routability in customization may cause design rule violation or yield problems. Designing a highly efficient layout of the wiring structure and high wire accessibility to allow greater flexibility in customization may introduce more vias and more difficulty in the customization stage.

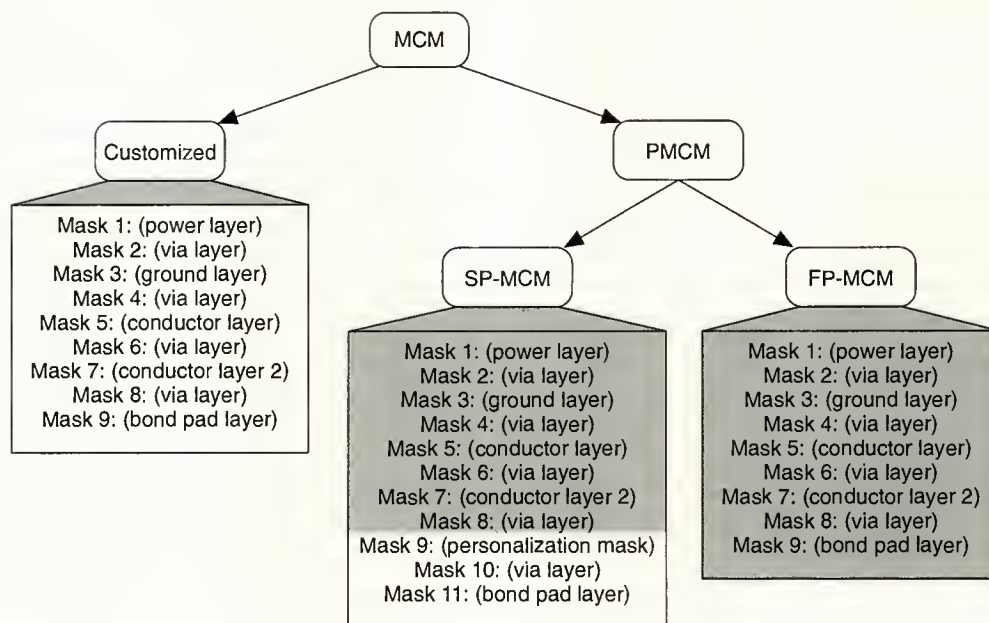


Figure 5. Comparison of manufacturing sequence for customized and QTAI high-density interconnections. Shaded regions indicate the generic substrate manufacturing steps.

Another important issue is electrical performance. High electrical performance in the final product is the ultimate goal in using the MCM packaging in the first place. However, the wide range of applications intended for the SP-MCMs makes the choice difficult. The general questions that concern designers include: How much performance can be "designed in?" What level of performance will satisfy most customers? How readily can electrical performance be scaled along with design rules? QTAI technology addresses signal delay more than signal fidelity since physical/electrical lengths of interconnections are relatively short in MCMs.

Another important issue concerning electrical performance is power distribution. In semicustom MCMs the power distribution is often predefined. It must be performance-driven as well as highly flexible. It also has to accommodate a variety of supply voltages and supply potentials as well as a variety of AC/DC current requirements.

Once the interconnections have been completed, bare chips need to be bonded. There are two options for bonding the bare chips. One places the bond pads on the third routing layer by avoiding the areas used for interconnecting terminals of these wire segments in the conductor layer and conductor layer 2. A separate layer for the bond pads is not necessary in this case.

Another way of bonding the bare chips is to add a separate bond-pad layer. In this case, bare chips can be placed

anywhere and can be bonded using wire bond, tape-automated bonding (TAB), flip-chip, or any combination of these techniques. The process of placing and bonding the chips in QTAI is then equivalent to one used in the fully customized MCMs.


QTAI has been tested in several actual MCM design. QTAI design routing is quite efficient when using the QTAI router developed by Carey.⁶ All his routing has exhibited less than 10 percent deviation, using half the perimeter of the smallest rectangle enclosing a net as an estimate. In addition to that, the router has demonstrated use of the available wires in excess of 65 percent. No significant routability penalty has been observed, relative to fully customized design.

QTAI substrate fabrication currently uses a CuPI process of a plated copper interconnection, stacked vias, and a planar dielectric. The process has implemented both customized and QTAI functional CuPI substrates. Users claim that the process readily supports the QTAI design rules of 15- μ m lines, 20- μ m minimum spaces, and 30- μ m via pads. Present substrates have lines on a 75- μ m grid (3 mil) that provide a substrate wiring density of approximately 670 in./in.² With reference plane metal added to the personalization layer, a balanced, dual-stripline construction results. Observers noted an impedance of 60 ohms and crosstalk of 2.5 percent for both *x* and *y* layers. Rise times of 200 ps were supported on 2-in. lines, and rise times of 500 ps were possible on 4-in.

lines. Users claim that the current QTAI technology supports clock rates of more than 100 MHz.

Future development is under way to incorporate thermal vias into the design to provide a thermal impedance through the multilayer interconnection well below $1^{\circ}\text{C}/\text{W}/\text{cm}^2$. Also under development is a low-cost laser-based tool usable for maskless personalization of QTAI substrates by the end user. The future trend for MCM design is higher performance for less cost and shorter turnaround time. QTAI, which combines the advantages of fully customized MCMs and FP-MCMs, proves to be a promising approach to solve the problems of high cost and long turnaround time for MCMs.

SEMICUSTOM OR PROGRAMMABLE MULTICHIP MODULES may play a key role in speeding up MCM development, acceptance, and application. The main advantage of PMCMs is that one generic layout can be used for implementing several different designs based on the user's requirements. The programmable feature of this type of MCM obviates the need to fabricate designs at the mask level offering fast turnaround time and high yield. Currently, electrical antifuse switches are used to program interconnections, however some manufacturers are also investigating the use of a laser-programmable switch.

PMCM technology reduces the cost of MCM development by a significant margin; it also degrades system performance as compared to fully customized MCMs due to electrical effects of the programmable elements. 

Acknowledgments

The National Science Foundation supported this research in part under grant USE-9052346.

References

1. R. Johnson, "Multichip Modules: Next-Generation Packages," *IEEE Spectrum*, Mar. 1990, pp. 34-48.
2. J. Mosley, "Lack of Standards Impedes Design Issues," *EDN*, Jan. 2, 1992, pp. 35-42.
3. T. Ormond, "Multichip Modules Expand Circuit Density," *EDN*, May 1991, pp. 67-76.
4. I. Dobbelaere et al., "Field-Programmable MCM Systems: Design of an Interconnection Frame," *Proc. FPGA Conf.*, 1992, pp. 52-56, 1992.
5. R. Berger, "A Laser-Programmable Multichip Module on Silicon," *Proc. IEEE Int'l Conf. Wafer-Scale Integration*, to be published.
6. D. Carey, "Programmable Multichip Module Technology," *Hybrid Circuit Technology*, Aug. 1991, pp. 25-29.
7. J. Banker, "Designing ASICs for Use with Multichip Modules," *Proc. IEEE Int'l Conf. Computer Design*, to be published.

8. R. Frankel et al., "SLASH—an RVLSI CAD System," *Proc. Int'l Conf. Wafer Scale Integration*, Jan. 1989.



Surendra Burman, a software engineer at Vista Technologies, Inc., is working on a project for electronic packaging. He received his BE degree in electronics and communications from Mangalore University, India, and his MS degree in computer science from Western Michigan University. He has published papers on VLSI design automation, FPGAs, and MCMs. He is a member of the International Society for Hybrid Microelectronics.



Naveed A. Sherwani is an associate professor in the Computer Science Department at Western Michigan University. His research interests include physical design automation of VLSI circuits, combinatorial optimization, and graph algorithms.

Sherwani received his MS and PhD degrees in computer science from the University of Nebraska-Lincoln. He has published over 50 refereed papers in various journals and conference proceedings, and his paper on three-layer over-the-cell routing received ICCAD-91's Distinguished Paper award. He has chaired four conferences, served on technical committees of several others, and is a member of the IEEE, the Computer Society, and the ACM.

Direct questions concerning this article to Naveed A. Sherwani, Dept. of Computer Science, Western Michigan University, Kalamazoo, MI 49008; sherwani@cs.wmich.edu.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 159

Medium 160

High 161



Interaction of Multichip Module Substrates with High-Density Connectors

Interfaces between multichip module substrates and high-density connectors experience mechanical stresses due to the simultaneous effects of mating forces and mismatches in the coefficients of thermal expansion. This work describes the stresses and strains in a substrate-conductor system and presents modeling results of experiments. Results show that definite relations exist between connector pitch, length, substrate thickness and material, and stresses in a connector body.

Yakov Belopolsky

*Berg Electronics
(E.I. du Pont de Nemours
and Company)*

A connector-substrate system experiences simultaneous multidirectional stresses caused by two prime sources: the difference in the material coefficients of thermal expansion and the mating forces. The interface between connector and substrate experiences the greatest structural loads in the system and, at the same time, often receives the least amount of attention from the multichip module (MCM) designer. Yet the designs may have to conform to demanding automotive applications that specify operating temperatures from -55 to $+100^{\circ}\text{C}$,¹ or military standards often requiring -65 to $+150^{\circ}\text{C}$. These high temperatures result in increased mechanical stresses on the connector-substrate interface.

Recently, many designers have viewed MCM technology as a temporary solution—a bridge between yesterday's PWB and tomorrow's ASIC. Lately however, due to design flexibility, power-handling capabilities, and a relatively short production cycle, MCM design has matured as a separate direction in electronic interconnection technology. Designers have used a variety of advanced multilayer substrate technologies in MCM interconnections: FR4, alumina and beryllia ce-

ramics, aluminum nitride, silicon, polyimides, very thin (0.2- to 1-mm) substrates, glass, insulated metals, and combinations of flexible and rigid materials. The choices in the substrate technology directly affect the second level of interconnection: a connector between the MCM proper and other parts of the system. Many advanced multichip designs use an area pin-grid array, others use elastomeric unsoldered connectors, while the traditional connector technology is often neglected.

The combination of high power, functional and physical densities, and electrical speeds dictates the connector design for MCM technology. Due to a great number of functions performed in a small space, MCMs have high operating temperatures and very thin conductors, and mechanical stresses become the prime cause of system failures. The problem is to identify and quantify the interface stresses so the system can be designed to reduce such stresses.

Experimental evaluation is a very long and expensive procedure. On the other hand, the application of computer technology, in particular, finite-element modeling, to connector design allows fast and cost-effective evaluation of new

MCM designs. This article attempts to help MCM designers reduce interface stresses by selecting the most suitable combination of connector body materials and substrate system.

Very high density connectors

A connector developer might feel the MCM system design trends are contradictory. The high signal-processing density in MCMs requires fewer interboard communications. However, the demands for a clear signal and line conditioning introduce a large number of ground connections, making the overall number of connections greater. A very significant increase in the number of system functions also results in a greater number of connections, so that the typical MCM design requires 400 to 800 connections.

The MCM is an island of very high speed signal processing in a slower environment. To achieve high density and speed, MCM designers use the most advanced interconnections (for example, 0.5- μm gate length CMOS and silicon-on-silicon interconnection technology). The fine-line interconnection requires a connection process with low mechanical stresses and fine pitch.

Initial MCM designs, based on the VLSI packaging approach, used traditional pin-grid arrays: multilayer, cofired, high-temperature ceramics with internal wire bonding. Obvious disadvantages limited that type of connection: high cost, problems with repairability, and difficulties in controlling electrical parameters, particularly inductance.

Later designs began using elastomeric connectors, which gained some popularity due to their fine pitch, relatively low cost, and ease of application. The attractive feature of elastomeric connectors is that terminals do not have to be soldered to the boards. Potential elastomeric disadvantages are inefficient power handling, susceptibility to contact resistance variations, and limited application temperatures.

The fast signal rise times of today's MCMs and PWBs³ require connectors of controlled characteristic impedance (in the 50-to-75-ohm range), low cross talk (1 to 2 percent at 500-ps rise times), and low inductance (less than 2 nanohenry). Also, a connector must be capable of multiple matings. For a connector designer, these requirements represent another contradiction: low inductance calls for a short contact path, while multiple mating requirements demand longer wiping distances.

The traditional metal beam in plastics connectors may meet all of the mature MCM design goals. These connectors allow a designer to use configurations, for example, edge card or straddle mount that were not possible with other connector schemes. Figure 1a shows a straddle-mount application of an electrically enhanced connector of high electrical performance and 0.5-mm equivalent pitch. To reduce cross talk and enhance electrical properties, its body is a combination of plastic and metal. Figure 1b shows a low-profile, fine-pitch board-to-board connector made of liquid crystal polymer. It

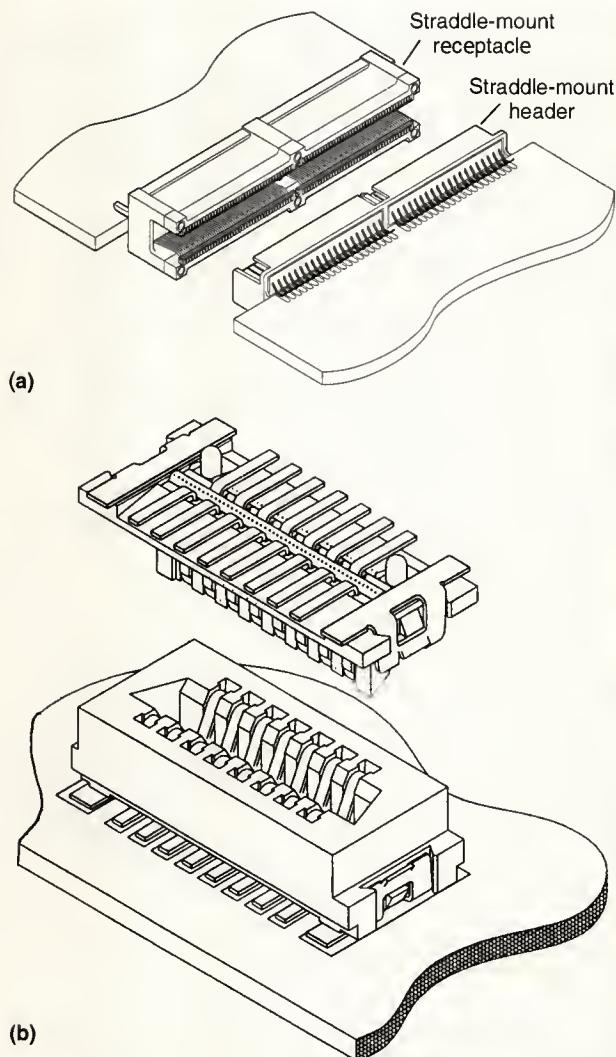


Figure 1. Very high density EMI-shielded connector (a)² and high-density, low-profile connector (b).

has a 0.5-mm equivalent pitch and a 4.15-mm mating height.

In this study my associates and I considered three material systems for connector bodies:

- liquid crystal polymer, or LCP ($E = 2.10$ gigapascals),
- polyphenylene sulfide, or PPS ($E = 2.70$ GPa), and
- plastic-metal combination, or PMC ($E = 3.60$ GPa),

where E is a modulus of elasticity.

In many cases, a connector influences the size of MCM substrates. Table 1, next page, illustrates the relations between connector pitch and its length.

Table 1. Length of a 100-I/O connector as a function of pitch, mm (in.).

Pitch		Connector length	
3.81	(0.150)	381	(15.0)
2.54	(0.100)	254	(10.0)
2.50	(0.098)	250	(9.84)
2.00	(0.079)	200	(7.84)
1.27	(0.050)	127	(5.00)
1.02	(0.040)	102	(4.00)
1.00	(0.039)	100	(3.94)
0.80	(0.032)	80	(3.14)
0.50	(0.020)	50	(1.96)
0.40	(0.016)	40	(1.58)
0.25	(0.010)	25	(0.98)
0.13	(0.005)	12.6	(0.50)

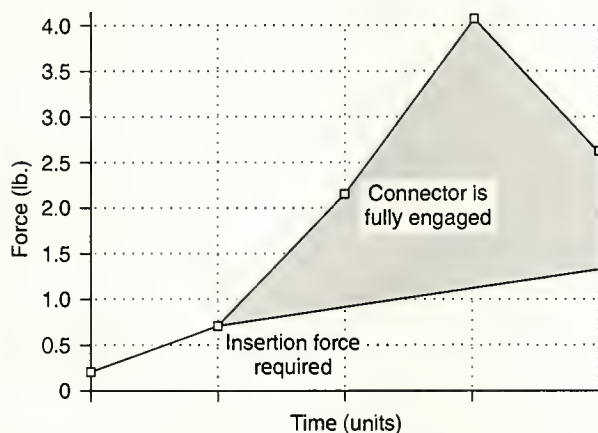


Figure 2. Insertion forces in blind mating (experimental data).



Figure 3. Multilayer MCM substrate of 0.635-mm alumina with eight 0.010-mm copper layers and 0.025-mm-thick polyimide dielectric.

Insertion forces, blind-mating problem

Connectors, unlike other electronic components, must withstand multiple applications of dynamic mating forces resulting from manual or automatic insertions. The manufacturer specifies the insertion forces, dependent upon the connector design and number of terminals. However, during manual insertion and field maintenance, users observed that the forces applied are often far in excess of the minimal required, particularly during a blind mating of two PWBs in portable electronic applications.

A simple experiment using several connector types with a different number of terminals showed that actual forces may exceed the minimum requirements by a factor of 4 (Figure 2). There was no clear relation between actual and specified forces in a case of blind mating. In fact, the force reflected experience of an assembly operator rather than a number of terminals.

Material systems and substrates

The properties of a plate element of a multilayer MCM substrate are defined by a flexural rigidity D :³

$$D = \frac{E}{(1 - \nu^2)} \int_{-t/2}^{t/2} z^2 dz$$

where t is the thickness of the layer, ν is Poisson's ratio, and z is the distance from a neutral axis.

For a rectangular plate, $D = Et^3/[12(1 - \nu^2)]$. Then, for a multilayer

$$D = (b/12)[(2n_1 + 1)t_1^3 E_1 + 2n_2 t_2^3 E_2 + (t_1 + t_2)^2 + [4n_1(n_1 + 1) * (2n_1 + 1)t_1 E_1 + (2n_2 - 1)2n_2(2n_2 + 1)t_2 E_2]]$$

where n_1 is the number of layers of material E_1 above the neutral axis, shown as a dashed line in Figure 3.

Tables 2 and 3 summarize the material and substrate data. The elastic modulus in Table 3 is, in fact, a weighted effective modulus of elasticity.

In the absence of constraints and outside forces, the substrate material under elevated temperatures tends to form a sphere of a radius:

$$r = t/q_{\text{eff}} * T$$

where, t and q_{eff} are the thickness and coefficient of thermal expansion of a complex multilayer board. T is the ambient temperature, in degrees Celsius.

In combination with pressure applied during insertion, a substrate deflects and stresses solder joints. The total deflection d is equal:

$$d = d_1 + d_2$$

Table 2. Properties of selected materials.⁴⁻⁷

Material	Thermal expansion coefficient (ppm/°C)	Thermal conductivity (W/m°C)	Elastic modulus (GPa)
Beryllia 99.5 percent	9.0	200.0	30-70
Aluminum nitride	4.6	140.0	45-60
Alumina 96 percent	8.1	18.0	50-60
Silicon	5.6	40.0	12-15
Aluminum	23.6	237.0	28.0
Copper	17.7	400.0	110.0
Solder 60/40 (Sn/Pb)	25.0	5.1	2.0
Teflon (PFA)	184.0	0.055	2.8
Kapton HN	31.0	0.155	2.6
Kapton MT	20.0	0.380	2.7
Glass (B-Si)	12.0	1.1	65.0
Epoxy-glass (FR4)	15.0	0.31	15.0
Liquid crystal polymer	25.0	0.7	2.1
Polyphenylene sulfide	3.0	0.8	2.7
Beryllium-copper alloy	16.0	180.0	122.0

Table 3. MCM substrate model list.

Base material thickness	Conductor thickness (mm)	Dielectric thickness (mm)	Number of layers	Overall thickness	E _{eff} (GPa)
Alumina 0.635	Copper 0.010	Polyimide 0.025	8	0.915	45
Alumina 1.020	Copper 0.010	Polyimide 0.025	8	1.300	45
AlN 0.635	W/gold 0.005	Polyimide 0.010	8	0.755	40
Polyimide 0.200	Copper 0.005	Polyimide 0.005	8	0.280	3
FR4 0.810	Copper 0.018	Epoxy 0.810	2	0.846	16
FR4 0.810	Copper 0.018	Epoxy 0.051	16	1.521	16
FR4 1.630	Copper 0.018	Polyimide 0.025	16	2.232	16
Silicon 0.200	Aluminum 0.005	Polyimide 0.010	8	0.320	13
BeO 0.635	Gold 0.010	Polyimide 0.025	8	0.915	50

where $d_1 = W^2/8 * r$ and $d_2 = f(P \text{ insertion})$

Here, d_2 is a function of the support location, material properties, thickness, and length of a connector.

Solder and solder joint

Despite (or because of) a very large number of literature sources dealing with solders and solder joints, we found that a variety of contradictory information exists on the basic properties of the eutectic solder alloy. We reduced the compilation of several sources into Table 4.

The anisotropic properties of solder material expressed as

a ratio of the shear yield stress versus tension yield stress (Poisson's ratio) equals 0.5. With computer modeling, we can simulate the solder joint defects in wide limits by manipulating the Poisson's ratio and yield stress. Figure 4 shows finite elements of an ideal solder joint.

Features of a high-density connector terminal. Three major characteristics of the MCM connectors related to the stresses in the surface-mount solder joints are

- a short cantilever beam,
- a composite cross section, and
- distributed loading.

All three of these features are present simultaneously in most designs. A connector terminal, best described as a cantilever beam, is significantly shorter now; 2 to 4 mm is the typical length. The wiping distances are less than 0.25 mm, and the terminal material is one of the brass alloys of a high (122 to 150 GPa) elastic modulus. Terminals are thin: 0.1 to 0.3 mm. The finished terminal has a rich history of thermal and mechanical conditioning: stamping and coining. The surface crystal lattice comprises a significant portion of the overall cross section, and the plating and soldering operations add other material layers to the overall width. The material properties of the processed terminals (the elastic modulus and yield strength) are different from that of the original material.

The load or applied multiaxial stress cannot be represented

Table 4. Properties of eutectic lead-tin solder (based on various sources⁶⁻⁸).

Property	Value/units
UNS no.	13360
Sn/Pb	63/37 percent
Melting temperature	182.7°C
Ultimate tensile strength	46 MPa
Coefficient of thermal expansion	24 ppm/°C
Elongation	50 percent
Yield strength	5
Hardness (Brinell)	12 MPa
Elastic modulus	2 GPa

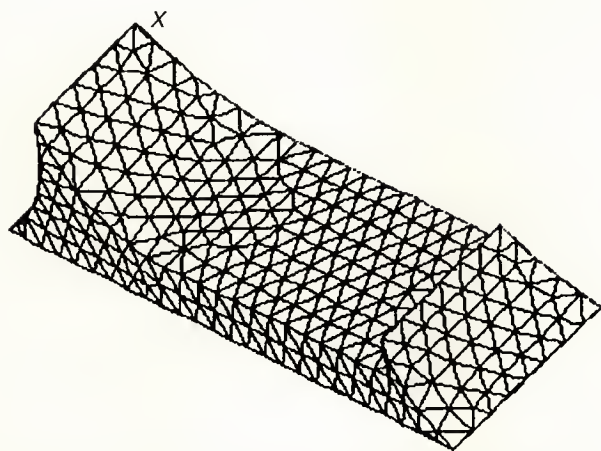


Figure 4. Finite elements of ideal solder joint.

by a single force vector, but becomes a complex distributed load. Note that the regions of the applied force vectors are very close and/or inside the solder joint. The most significant result of this change is that the solder joint becomes a part of a composite cantilever beam of variable geometry. Figure 5 illustrates loads in a surface-mount technology terminal due to multiaxial stresses.

Strains resulting from temperature effects. In general, we see two types of strains: thermal expansion and those created by temperature stresses that occur when two restrained bodies with different CTEs (coefficients of thermal expansion) expand or contract, and are attached to each other. The strain in one body results in the stress and corresponding additional strain in another body. As a necessary simplification, we assumed we had eight stress areas created by a CTE mismatch (see Table 5).

A significant body of experimental data in the literature describes the effect of a CTE mismatch on solder joint design. Obviously, the experimental data describes the summary effect of the stresses listed in Table 5, as well as other variable causes such as solder properties, joint geometry, and so on. Assume that the stresses due to solder CTE mismatches to all other materials are applied to the solder region only, the operating temperature is 50°C, and the elastic moduli are constant in the operating range. Then, the following conditions apply:

- no solder or other parts of the connector were restrained during formation of the solder joint,
- the linear dimensions were equal at 25°C,

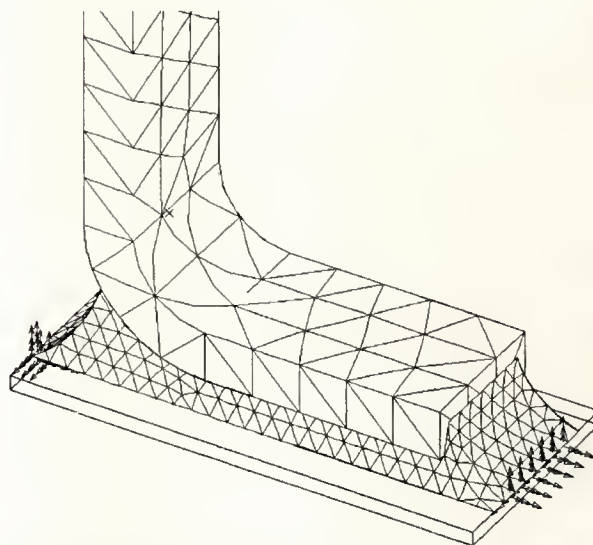


Figure 5. Gull-wing joint loads.

- the formation stage occurred at ambient temperature 200°C, and
- instantaneous stress (no creep) is assumed for the temperature rise from 25 to 50°C.

Table 6 shows the calculated unrestrained and differential dimension changes. The terminal material constricted the expansion of the solder, resulting in applied stress.

The resulting strain causing stress $s_{res} = s_{max} - s_{min}$ where s_{max} and s_{min} are strains of the adjacent surfaces. So that

$$s_{res} = u * T * (q_1 - q_2)$$

where u is a length of solder along an interface.

Expansion of connector body

The body of an MCM connector shown in Figure 6 will expand due to elevated temperatures in all directions. In the worst case, we consider a connector body unconstrained, so that the total expansion is applied as strain to the solder joint. Neglecting the expansion in the z axis, the expansion of the

connector body in x direction at distance L_0 from the neutral axis occurs at 50°C:

$$L_{50} = p(n + 0.5)q_b * \Delta T$$

Here, p is the effective pitch in mm, n is the terminal position, and q_b and q_s are CTE ppm/°C of materials of the connector body and substrate. The dimension L_0 is, in this case, $L_0 - p(n + 0.5)$. Shear stress $\tau = G * y_{eff}$, where G is a modulus of rigidity, and $G = E/[2(1 + \nu)]$.

Now, $t_j = t_b - t_s$, where t_b and t_s are thermal stresses (not shear stresses) in a connector body and a substrate material. So, where l is a linear dimension along the solder joint interface

$$\begin{aligned} y_{eff} &= l_b - l_s \\ l_b &= q_b * L_0 \Delta T \\ l_s &= q_s * L_0 \Delta T \text{ and} \\ y_{eff} &= (q_b - q_s) * L_0 \Delta T \\ t_b &= G_b(q_b - q_s) * \Delta T \end{aligned}$$

Finally, $t_j = (G_b - G_s) * (q_b - q_s) * \Delta T$.

Table 5. CTE mismatch areas.		
No.	Material 1	Material 2
1	Con body	Substrate
2	Con body	Terminal
3	Solder	Terminal
4	Solder	Substrate
5	Solder	Metallization
6	Solder	Substrate
7	Con body	Metallization
8	Metal	Substrate

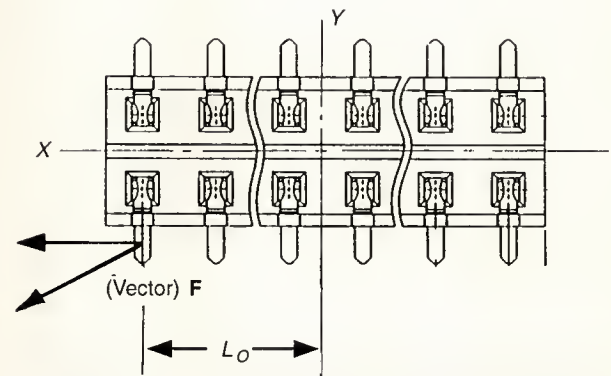
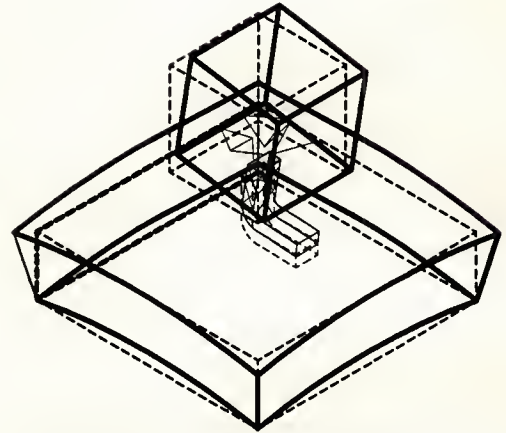
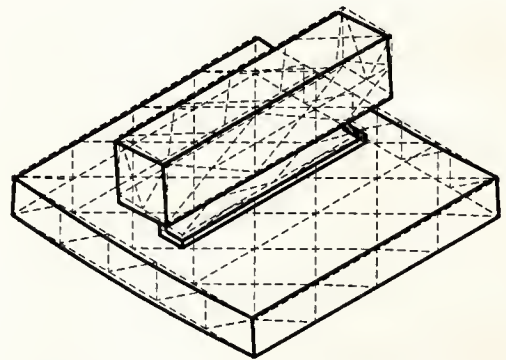


Figure 6. Asymmetrical expansion of connector body.

Table 6. Thermal strains in solder joint.				
Item	Unrestricted length, mm			Differential strain (mm) @50°C
	Design @25°C	Formation @200°C	Operating @50°C	
Terminal	1.0	1.00280	1.00040	-0.00022
Pad	1.0	1.00315	1.00045	-0.00017
Solder	1.0	1.00438	1.00062	0
Substrate	1.0	1.00260	1.00038	-0.00024

Table 7. Combined stress experienced by solder joint.

Substrate (mm thick)	Connector body	Stress factor (%)	Percent of maxi- mum yield stress
FR4 (0.846)	LCP	100	25.0
	PPS	110	27.5
	MPC	130	33.0
Alumina (0.915)	LCP	138	34.5
	PPS	67	16.7
	MPC	60	13.0
Aluminum Nitride (0.915)	LCP	180	45.0
	PPS	77	19.3
	MPC	56	16.5
Silicon (0.320)	PPS	130	33.0
	MPS	120	30.5
Beryllia (0.915)	PPS	150	38.0
	MPS	138	35.0
Polyimide (0.280)	LCP	64	16.0
	PPS	176	44.0
	MPC	190	60.0

**Figure 7. Displacements under insertion load.****Figure 8. Displacements under combined load in a polyimide substrate.**

Obviously, the maximum stress in a solder joint

$$t_{\max ys} = K * t_{ys}$$

where K reflects the anisotropic properties of solder, safety margin, solder joint geometry, and defects.

$$K = (0.2 \text{ to } 0.5) \text{ and}$$

$$t_{ys} = 35 \text{ MPa; also,}$$

$$t_{\max ys} = F * a * b$$

F is the force acting on a joint, and a and b are dimensions of a rectangular cross section.

We considered the forces caused by expansion of a plastic body to have two components: The first acts along the length of a connector, and the second is perpendicular to it. Both of these components can be presented as loads:

$$F_i = (G_b - G_s) * (q_b - q_s) * \Delta T * a * b$$

where q_b and q_s are thermal expansion coefficients of a sub-

strate and plastic material. Thus, we can express it as

$$P_{\text{sum}} = (P_1^2 + P_2^2)^{0.5}$$

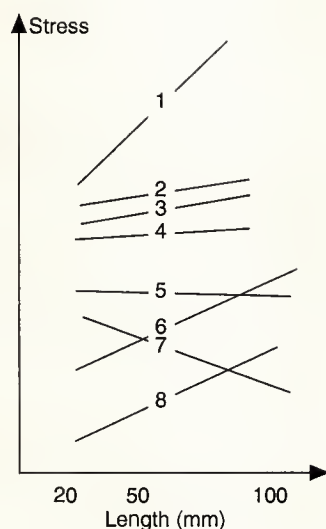
We define the force vector as

$$\tan(\alpha) = 2 * l_{n \text{ lead}} / w$$

where w is the connector width.

Modeling results

In combined stress models we applied simultaneous stresses as structural loads. We performed the analysis using the IBM CAEDS mechanical finite-element analysis and modeling software package. We created four model families: composite (lead, solder, metallization, and substrate), lead, solder, and connector body. We used solid trapezoidal elements, and the computer created a mesh with 2,460 to 8,230 elements; element distortion did not exceed 20 percent. Table 7 lists the modeling results. In this table the stress factor equals a



(a)

No.	Substrate (mm thick)		Connector body
1	AlN	(0.915)	LCP
2	FR4	(1.63)	MPC
3	Silicon	(0.320)	MPC
4	FR4	(1.63)	LCP
5	FR4	(0.846)	LCP
6	Alumina	(1.30)	PPS
7	Polyimide	(0.280)	LCP
8	Alumina	(0.915)	PPS

(b)

Figure 9. Stresses in connector body normalized to 0.846-mm-thick FR4 with an LCP connector (a) versus substrate composition (b).

normalized stress according to the formula:

$$S_i/S_p * 100 \text{ percent}$$


where S_p is the stress in 100 I/O 0.5-mm pitch connectors made of LCP on a 0.846-mm-thick FR4 substrate.

Figures 7 and 8 show the stresses and displacements in terminal and connector. Figure 9 illustrates the relations between stresses in a connector body as a function of connector length, substrate material, and thickness.

USING MINIATURE SURFACE-MOUNT CONNECTORS, we experimentally verified the mating forces and found that they significantly exceed that minimally required by a connector design. We combined the stresses created by blind insertion with stresses introduced by the CTE mismatch in the solder joint due to nondissipated temperature excursion stresses. A combined model included stresses related to the thermal expansion of a connector body. The results showed that the choice of materials for a substrate and connector body dramatically affect the system stresses. A combination of a polyimide substrate and LCP resulted in the lowest stress on connector terminals.

On the other hand, a high CTE LCP material created almost three times higher stresses on the aluminum nitride substrates. The stresses in a connector body are directly proportional to its length at a given ambient temperature. Re-

duction in the connector pitch results in reduced solder joint stresses. The stress on a connector body was lower on the thinner substrates. An analytical approach allows evaluation of a wide range of connector designs, solder joint, and substrate materials. New high-density surface-mount connectors must be compatible with novel material technologies and fine-pitch assembly processes.

Finite-element modeling and analysis together with experimental studies proves to be the best method to produce new good-quality designs fast and cost effectively. We hope the conclusions presented here will assist an MCM developer in reducing the interface stresses and improving the system reliability by selecting the most suitable combination of an interconnection substrate and the connector body materials. 

Acknowledgments

I thank P. Curwen and T. Lyons for their continuous support and helpful discussions, and T. Lemke and A. Raistrick for assistance with graphic information.

References

1. Y. Belopolsky and C. Killian, "Flexible Materials for Automotive Electronics Applications," *Proc. Int'l Symp. Automotive Technology and Automation*, Automotive Automation Ltd., 42 Lloyd Ave.,

New Books

GENETIC ALGORITHMS

edited by Bill P. Buckles and Frederick E. Petry

The first three papers in *Genetic Algorithms* discuss a framework for an overall classification of GAs based on parameters and strategies and examine issues such as optimization techniques, implicit parallelism, and the sizing of populations. The next three deal with critical aspects of genetic algorithms and learning, including parameter-based techniques and the topic of classifier systems. The last group of articles cover the application of genetic algorithms to pattern recognition, routing and permutation problems, and neural networks.

Articles Included: An Overview of Genetic Algorithms and Their Applications, Optimization of Control Parameters for Genetic Algorithms, How Genetic Algorithms Work, Learning with Genetic Algorithms, Using Genetic Algorithms for Supervised Concept Learning.

120 pages. July 1992. Softcover. ISBN 0-8186-2935-5.
Catalog # 2935-05 \$35.00 Members \$25.00

To order call toll-free
1-800-CS-BOOKS
or 714/ 821-8380

 IEEE COMPUTER SOCIETY

 THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.

IEEE COMPUTER SOCIETY PRESS

DIGITAL IMAGE PROCESSING, 2nd Edition

edited by Rama Chellappa

This collection of papers examining digital image processing is a revised and updated version of the first edition, published in 1985. It concentrates on discussions of image compression, explorations into image enhancement with an emphasis on median and related non-linear filtering of images, and investigations of image restoration to preserve discontinuities while smoothing or restoring.

The tutorial includes an overview of all five sections and a total of 47 papers of which 35 are new to this edition. The text explores models for:

- | | |
|----------------------------------|-------------------------------|
| * Image Data | * Filters in Image Processing |
| * Adaptive Restoration of Images | * Image Coding Techniques |
| * Vector Quantization | * Image Sequence Compression |

Sections: Image Models, Image Enhancement, Image Restoration, Image Data Compression, Emerging Topics, Bibliography.

816 PAGES. APRIL 1992. HARDCOVER. ISBN 0-8186-2362-4.
CATALOG # 2362-01 — \$80.00 MEMBERS \$64.00



To order call: 1-800-CS-BOOKS
or FAX 714-821-4010

Croydon, England, May 1991, pp. 431-438.

2. Berg Electronics (Du Pont Connector Systems), *Catalog A*, Part 3-1, Feb. 1992.
3. Y. Belopolsky, A. Murphy, and E. Abramson, "Precision Controlled Impedance Multilayer Printed Wiring Boards with Permanent Solder Masks," *Proc. Int'l Microelectronic Conf. Tokyo*, Society for Hybrid Microelectronics, Kodairo, Tokyo 18, Japan, June 1991, pp. 540-544.
4. Y. Belopolsky and T. Poulin, "Thermal and Reliability Modeling of Thick Film and Other Material Systems for Under-the-Hood Automotive Applications," *Proc. Int'l Microelectronic Conf. Tokyo*, Society for Hybrid Microelectronics, Kodaira, Tokyo 18, Japan, June 1990, p. 363.
5. *Metals Handbook*, sixth ed., H.E. Boyer and T.L. Gall, eds., American Society for Metals, Metals Park, Ohio, May 1991, pp. 10-2, 12-3.
6. P.J. Martin et al., "Flexible Fluoropolymer Laminates," *Proc. First Electronic Materials and Processing Congress*, American Society for Metals, Sept. 1988, p. 367.
7. *Microelectronic Packaging Handbook*, R.R. Tummala and E.J. Rymaszewski, eds., Van Nostrand Reinhold, New York, 1989, pp. 669, 741.
8. J.H. Faupel and F.D. Fisher, *Engineering Design*, John Wiley and Sons, New York, 1981, pp. 209-210, 308-311.



Yakov Belopolsky is a senior research associate with Berg Electronics of E.I. du Pont de Nemours and Company. He works on applications of computer modeling and finite-element analysis to thermal, mechanical, and electrical problems.

Belopolsky holds MS and PhD degrees in solid-state physics and mechanical engineering. He is a registered professional engineer, an associate of the World Space Foundation, and a member of the IEEE, the International Society for Hybrid Microelectronics, and the Society for Automotive Engineers.

Direct questions concerning this article to the author at Berg Electronics (E.I. du Pont de Nemours and Company), 825 Old Trail Road, Etters, PA 17319.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 162

Medium 163

High 164

NEW BOOKS

from **IEEE Computer Society Press**

VISUAL CUES: Practical Data Visualization

by Peter R. Keller and Mary M. Keller

Visual Cues is a user-friendly, practical book about scientific visualization. It is written primarily for scientists and engineers confronted with the challenge of determining how to visualize data. The first four chapters in this new monograph present general information about producing an effective image. They describe a basic methodology for image construction, discuss various means for viewing computer-generated images, explain general design principles for creating meaningful, intelligible images, and cover rules for effectively using color.

Chapter 5, the heart of the text, presents by example techniques for rendering visible information hidden in data. The chapter discusses what information is revealed by use of a specific technique and demonstrates the application of the technique in the accompanying image. References to additional literature are also provided. Although the 150 images are from a broad range of disciplines, the discussion of the techniques is presented independent of discipline to show readers that many visualization techniques can be generally applied.

To help the scientist and engineer relate particular visualization needs to the examples in *Visual Cues*, the authors display certain information for each example:

- * A running head identifies images according to general ways of thinking about data.
- * A 3-line index gives the number of variables in the image, describes the information revealed, and names the application.
- * A succinct image caption indicates the techniques and the information revealed by the technique.

The text accompanying each image describes the application more fully, names the variables, and lists the hardware and software used to produce the image. An additional paragraph discusses the techniques that reveal the information. Other images containing related information are cross-referenced, and each example is evaluated with respect to the actual amount of computer power needed to accomplish the visualization.

Sections: Introduction, Data Visualization, Output Media, Image Design, Color, Illustrated Techniques, Appendices.

240 pages. December 1992.
Hardcover. ISBN 0-8186-3102-3
Catalog # 3102-04 — \$79.95 Members \$64.00

Call toll-free to order: 1-800-CS-BOOKS

or order by FAX: (714) 821-4010

(in California call (714) 821-8380)





Amalgams for Improved Electronics Interconnection

Amalgam systems offer a promising alternative to traditional electronic solders for lower temperature, hence lower cost processes and components, without the environmental drawbacks of most solder systems. Amalgams are nonequilibrium, mechanically alloyed materials formed at or near room temperature between a liquid metal and a powder. They offer exceptional thermal stability and superior joint strength and thermal cycle measurements.

Colin A. MacKay

Microelectronics and
Computer Technology
Corporation

From the earliest stages of electrical technology development to the most recent microelectronics advances, solders have been the material of choice for assembling components and assemblies. However, over the years, users have encountered certain drawbacks or limitations to their use. For example, given their tendency to creep or distort even under low loads, solders ideally should be used only in no- or low-stress applications. Also, fatigue behavior limits their use in cyclic mechanical or thermal environments.

Other limitations arise from the very nature of the soldering process. Because the solder must be melted to achieve wetting, hence bonding, the assembly automatically experiences a temperature excursion of 200°C to 300°C, depending upon the materials selected. Thermal excursions lead in turn to distortion and stresses as the materials are heated and cooled. The massive designs and coarse intercomponent spacings of older assemblies tolerated these distortions, but not so today's finer spacings and smaller components. Previously tolerable pad movements now create serious alignment problems. And recently, environmental considerations, particularly the need to minimize the amount of lead in materials, have begun to influence solder alloy choices as much as problems with the materials themselves.

Recent studies with amalgam systems using liquid metals and powders in a dental amalgam for-

mat show that these materials can be used for microelectronics bonding.¹⁻⁶ Although formed at or near room temperature, they have exceptional thermal stability, with melting points that range from near 300°C to above 600°C, depending on the system used. Joint strength and thermal cycle measurements show superior strengths and performance to solders. As with any emerging technology, not all problems have been solved. For example, the lengthy curing time required with the few systems studied precludes consideration of an in-line process. Still, these materials have been successfully applied to die-attachment (die up to 1.00-inch square), flip-chip, and chip-on-glass procedures, which we describe later on.

Amalgam characteristics and advantages

As fluid, mechanically alloyed mixes of liquid metal with a powder, amalgams initially form at room temperature as mobile metal fluids. In this state they offer unique advantages over solders whether they are plated, reflowed, or used as "hot dipped" coatings. Assembly to the liquid metal bonding condition occurs at room temperature, greatly reducing equipment complexity and cost and increasing process flexibility.

Their unique physical properties also impart special benefits. Amalgams are completely conformable, can accommodate large out-of-plane engineering tolerances, and can be dispensed or

printed precisely at the bond position. Further, amalgams can be tested in the liquid state, thus allowing electrical testing, circuit balancing, and matching or replacement of faulty components and devices before the final assembly stage, when the bond is cured.

Because of their inherent high thermal stability, amalgams can be processed at high temperatures after curing for such assembly stages as hermetic sealing and possibly even glass-to-metal sealing. They can attach difficult, geometrically shaped components as the final assembly process stage even though the components might serve a thermal function, for example, heat sinks. This ability could eliminate the need for complex and expensive jiggling throughout the assembly line. Also, amalgams can serve as precise alignment materials for such things as optic fibers in optoelectronic devices that require subsequent assembly steps. For example, hermetic sealing commonly causes misalignment of such precision devices with conventional solders.

The liquid metal constituent of the systems studied readily wets a range of ceramic and dielectric materials directly without any requirement for surface metallization. This property imparts further unique opportunities for amalgam applications:

- fewer process stages are needed,
- structural applications are available both for 3D substrates or sequential build-interconnect-build processes, and
- a wider range of materials can be used, either as ceramics, difficult-to-wet metals, or combinations of these.

Potential applications

As we shall see, amalgams offer possible advantages to a wide range of emerging technologies.

Die attachment. As die become larger, achieving a continuous, pore-free bond to a chip carrier or substrate becomes more difficult. So does accurately positioning die contact pads relative to the bonding pads of the chip carrier or substrate and assuring long-term reliability of the die attachment. For solder die attachment the level of stress in the die resulting from the soldering temperature thermal excursion or thermal cycling during operation increases with die size. And yet increasing die size is the natural trend for increasing functionality on die.

An obvious way to combat the stress problem is to reduce the magnitude of the temperature needed to attach the die. With their at or near room temperature processing, amalgams give exactly that control. Preliminary experiments show that silicon die at least up to 1.00-inch square can be bonded to alumina with minimal stress or distortion. They will also survive thermal cycle testing between -65°C and $+150^{\circ}\text{C}$ for more than 3,000 cycles. We have not yet determined the upper chip size limit for this material, so the possibility remains that it could be used to bond solar cells.

Ready handling of amalgams at room temperature eases addition of material to the bond zone. Also, X-ray examination of bonded chips shows that porosities at 1 percent or less are simple to maintain. The capability to move and adjust chip position at will while the amalgam is fluid, coupled with the benign curing temperatures that cause minimum movement, greatly reduce alignment problems.

Hermetic sealing. Multichip module technology often requires hermetic sealing of assemblies 3-, 4-, and up to 6-inches square, presenting a severe challenge given the temperatures that standard soldering techniques require. With modules this large and because of the design and materials of the package involved, distortion and bowing arise that in turn create stresses and solder conformability problems. Such difficulties can make this a low yield process, one requiring extensive reworking. Other than the difficulty of actually sealing the package, problems emerge when the package contains an assembly with precise component alignment requirements, such as an optoelectronic device. At the low temperatures available with amalgams, alignment tolerances are more readily maintained.

Another potential benefit of using amalgams for hermetic sealing is that ceramic seal lids can replace metal lids for chip carriers. Amalgams permit the elimination of metallizations because of their direct wetting capability on ceramics. Also, cheap, unpolished standard ceramic substrates can be used for the seal lid because of the conformability of amalgams.

Flip-chip attachment. Flip-chip assembly requires elevated contact areas on the chips to raise the chip surface above the substrate. In the IBM C4 system, the nearly hemispherical 10-percent tin/90-percent lead solder bumps provide the raised surfaces. The bumps themselves are formed either by "hot dipping" in a molten solder bath or by melting alloy deposits that have been applied by evaporation. Other systems use bumps made of solder, gold, or gold-plated nickel that are applied to either the substrate conductors or to the chip I/O.

Whatever the system used, this requirement for raised contact areas (bumps) is a disadvantage for the flip-chip method: The bump placement entails one or more extra processing stages. Assembly steps thus must be dedicated to a flip-chip approach because bumping is not a discretionary operation that can be applied at will within a flexible assembly process. The bumps must be added either at the wafer fabrication stage or during bulk substrate manufacture.

Experiments using amalgam as both the stand-off bump material and the bonding alloy demonstrate that silicon devices can be flip-chip mounted with 3-mil-diameter contact pads onto 3-mil-wide conductor pads with a maintained stand-off of 1 mil. These experiments involve a discretionary, software-driven process that can sequentially assemble a large variety of different chips, whatever the circuit design required.

Chip on glass. With the advent of portable computers,

particularly notebook and palm-top models, the volume occupied by the hardware is becoming increasingly important. The display screen that takes up so much volume is becoming more and more attractive as a substrate to support some of the electronics functions. The back of this display or its nonvisible perimeters can carry electronics components provided that the components adequately adhere to conductor tracks on the glass. This is the essence of the chip-on-glass technique. The substrate used in the flip-chip demonstration described earlier was glass with copper conductor tracks, effectively demonstrating this capability.

Lead-free solders. Ground water leaches lead out of landfills into our drinking water. Proposed environmental legislation seeks to reduce the amount of lead in manufactured goods that find their way into those landfills. Since lead in solders is a prospective target for such legislation, we must find materials that can replace lead-containing solders and devise solutions to problems arising from soldering with these substitute materials.

Amalgams are just such lead-free materials. None of their constituents carry the toxic hazard of lead; gallium is nontoxic. These materials achieve excellent electrical contact and mechanical adhesion with demonstrated elevated temperature and thermal cycle performance. A variety of bonding applications have proven their strength. Now we must examine the processes and requirements for such applications as surface-mount technology and through-hole bonding.

Optical interconnection. Early experiments indicate that a deposit of amalgam fluid suitably applied may be used to hold an optic fiber while it is aligned using a laser source. Then, after a laser burst stiffens the metal sufficiently to hold the fiber in position, subsequent steps may complete the optoelectronic device assembly without misaligning the optic fiber.

Automobile applications. The severe under-the-hood conditions of some automobile applications require extremely good, high-temperature performance. Certain regimes recommend testing up to $\sim 180^{\circ}\text{C}$. Classical solder phase diagrams have a 185°C horizontal tie line from near 100-percent tin to 19-percent tin. All compositions in this range therefore have some remnant of this low melting point component in their microstructures, which drastically reduces their elevated temperature creep strength. Amalgams, though formed at low temperatures, have high melting point constituents, thereby offering a potential solution to some of the current problems arising from the limitations of existing solders. Work studying the use of amalgams⁶ for die attachment, for example, shows that amalgams have exceptional thermal cycle performance up to at least 150°C . Since they have no low melting point components in their microstructure, 180°C seems within reach.

Heat-sink attachment. Typically, heat-sinked devices are built by first metallizing the substrate and brazing, in the case

of a ceramic, or bonding the heat sink with a high-temperature solder. This first operation is performed at the highest assembly temperature. No subsequent assembly stage then will disturb or be disturbed by the assembly sequence. If good thermal interfacing or precise positional tolerances are necessary at particular stages in the assembly operation, the complexly shaped casting or die casting then is fitted into a complexly shaped jig. This jig then transports the assembly through the subsequent assembly stages. Such jigging is often complicated and expensive. The high post-cure thermal stability of amalgams allied with the low curing temperature allows the heat-sink attachment operation with these materials to come last in an assembly process since near room temperatures can be used during the attachment. Putting the heat-sink attachment last eliminates complex, expensive jigging throughout the assembly line.

Tape-automated bonding for inner-lead bonding. The next generation of TAB is likely to have an array of I/O pads covering the total area of the top of the die. This will improve silicon utilization and I/O density. Consequently, an interconnection method will be needed that is benign to the problems of connecting to I/O pads situated over active devices on a chip. The liquid, being a highly conformable state requiring very low bonding pressures, offers this possibility with amalgams.

Via filling. For double-sided ceramic circuits, the via holes in ceramic substrates must be completely filled with a conductive medium, especially if unmetallized vias are used. This via filling in itself represents a significant problem. If, additionally, the filling must be hermetic, so that the circuits can themselves be used as a seal lid, we can place components requiring hermetic protection on one side of the substrate and those that do not on the other. Assembly problems then are compounded. Amalgams that will wet the surfaces with or without metallization offer real solutions to these problems. Simple, single-screening or dispensing operations can replace complex, multistage screening or tenting operations, with significant savings in time and equipment.

Structural bonding. A material that requires no surface metallization and that can be processed at or near room temperature would simplify applications such as mother- or daughterboard assembly. Using amalgams as structural materials permits such 3D electronics developments as memory cube, memory block devices, or simply 3D alumina or other materials geometries. (In memory cube, a block of silicon memory chips are physically bonded together to create a solid cubic structure which is sometimes called a memory block. In 3D alumina, double-sided alumina circuits are bonded together to form a solid 3D structure.)

Other applications no doubt exist, limited only by the innovation of the designer. Examples might include temporary interconnection design breadboards and temporary test interconnection for defect analysis and malfunction repair.

Table 1. Some possible amalgam systems.

Liquids (melting point, °C)	Powders
Mercury (-39)	Antimony
Gallium (30)	Cobalt
Indium (159)	Copper
Gallium/Tin (16)	Chromium
Gallium/Indium (15)	Germanium
Gallium/Indium/Tin (5)	Gold
Other combinations with Hg, Cd, Bi	Iron
	Nickel
	Magnesium
	Manganese
	Platinum
	Palladium
	Silver
	Vanadium

About amalgam systems

Amalgams are nonequilibrium, mechanically alloyed materials formed between a liquid metal and a powder. As such, they exist in a large number of systems. Table 1 lists some of the possible combinations of liquid elements, alloys, and powders that form amalgams.

As with any metal alloy system, amalgams can be formulated in binary and multicomponent systems. They offer a complexity beyond that of solders where the most commonly used alloys are based upon variations and modifications to the tin/lead system.

Systems studied. Amalgams can be formed in a wide variety of systems.⁷⁻¹² To be useful for microelectronic bonding, they must be capable of being mechanically alloyed (amalgamated) long enough to allow reproducible processing. That processing must take place under conditions that will initiate the mechanical alloying necessary to produce a material that will harden at low temperatures. The formulation's composition must contain enough powder component that it will set up and cure to form a solid, but not so much that it is solid in the amalgamator capsule or so stiff that it sets up in a few minutes. (See Figure 1 for set-up time and curing time definitions.) A range of compositions for each system meets this requirement.

Gallium/nickel forms usable amalgams in the range of 30- to 45-percent nickel. When fully solid these have microstructures as shown in Figure 2, which shows remnant particles of nickel embedded in a NiGa₄ reaction product. In general, the materials with lower nickel content take longer to set up and cure than those with higher contents. Table 2, next page, shows the effect of the curing temperature on the curing time.

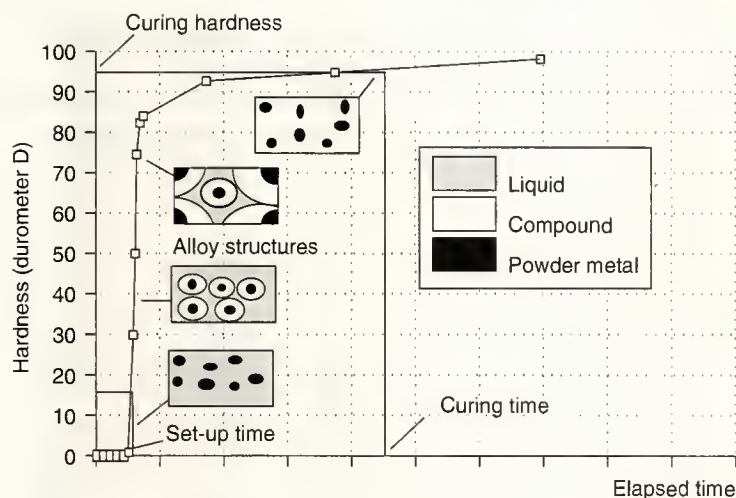


Figure 1. Hardening profile and microstructure of amalgams.



Figure 2. Microstructure of solid gallium/nickel amalgams.

In the course of studying this material under thermal cycle conditions, we found a phase transform not indicated by the phase diagram. At 120°C, we observed a solid-to-solid transformation of the Ni/Ga compound from NiGa₄ to a mix of NiGa₄ plus Ni₂Ga₃ with an associated 25-percent increase in volume. The volume increase shattered the bond interface, disrupting the joint.

Gallium/copper forms usable amalgams in the composition range of 25-percent to 40-percent copper (see Table 3).

Table 2. Set-up and curing times for Ga/Ni amalgams at various temperatures.

Amalgam	Set-up time (minutes)	Curing time (minutes)				
		35°C	45°C	60°C	100°C	150°C
30% Ni	575	~4,500	1,000		135	
45% Ni	50	1,250	245	190	55	~40

Table 3. Set-up and curing times for Ga/Cu amalgams at various temperatures.

Amalgam	Set-up time (minutes)	Curing time (minutes)			
		35°C	50°C	85°C	100°C
25% Cu	1,500	12,000	8,000	6,300	6,000
40% Cu	50	6,000	2,300	1,300	1,000

The micrographic structure is similar to that for gallium/nickel but with the matrix compound being CuGa_2 . See Figure 3 for a micrograph of a gallium/copper amalgam. Variations of physical properties follow similar rules to those for nickel, but with vastly extended set-up and curing times.

Thermal cycle experiments with these alloys showed very good performance with no phase change problems. Joint strengths were good but the copper-containing materials were more difficult to wet (about equivalent to pure gallium) than were the nickel-containing alloys.

Gallium/copper/nickel alloys could be formed for powder percentages between 26 and 50 percent of nickel, plus cop-

per mixes blended in ratios from 0.111 Ni:Cu to 0.429 Ni:Cu. These correspond to compositions of 3.3-percent Ni/26.7-percent Cu/Ga to 22.9-percent Ni/30.5-percent Cu/Ga. Figure 4 shows a typical micrograph of a gallium/copper/nickel amalgam. Table 4 shows the range of compositions that exhibit suitable amalgamation properties, while Tables 5 and 6 show typical set-up and curing (hardening) conditions.

From these alloy compositions we selected a 5-percent Ni/30-percent Cu/65-percent Ga composition as standard. Table 7, next page, gives a typical range of materials properties in this system.

Gallium/silver experiments were not entirely successful because silver powder was not available in a sufficiently coarse powder grade. The fine silver reacted completely and too quickly, yield-

ing an equilibrium structure with all the powder dissolved as shown in Figure 5. Consequently, we could not produce a liquid amalgam that while initially fluid would subsequently cure to full hardness.

We have not yet completely examined gallium/copper/silver and gallium/nickel/silver combinations. Using the same methodology as for nickel/copper mixes produced the results shown in Tables 8, 9, and 10.

Amalgamation methods. Amalgam components may be mechanically alloyed in a variety of ways; by hand mixing, by beating in a blender, or by inserting an ultrasonic probe into a dispensed mix of the liquid and powder parts. In our experi-



Figure 3. Micrograph of solid gallium/copper amalgam (X500).

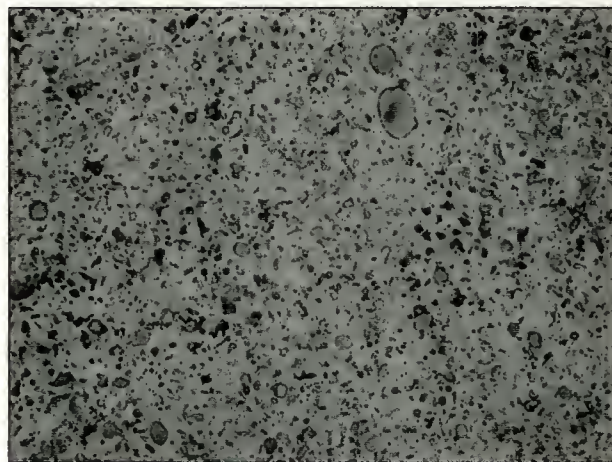


Figure 4. SEM of solid gallium/copper/nickel amalgam (X100).

ments, we used a commercial dental amalgamator consisting of a capsule containing a pestle used to hammer the powder in the presence of the liquid. See Figure 6 (on page 53).

Requirements for electronics amalgams. The physical requirements for a usable amalgam in dentistry differ from those needed to use these materials for bonding electronics. A dental amalgam works by filling the cavity, or caries, in a tooth with the amalgam. On solidification, the amalgam swells slightly to lock the cohesive mass of the metal into place. By comparison, electronic joints must wet and join two flat or coaxial surfaces. For the comfort and convenience of patients dental amalgams must set up in a few minutes to a sufficient stiffness to prevent damage, and must harden in under 2 hours at body temperature. In joining two flat surfaces for electronics, the amalgams must be more fluid to allow wetting of the joint surfaces. This fluidity must persist at room temperature long enough to offer a reasonable bench life that gives a sensible processing window. Yet, to retain the low-temperature benefits of an amalgam, only slightly elevated temperatures should be used to accelerate the hardening process.

Amalgam hardening mechanism.

Figure 1 showed a typical hardness versus time curve for an amalgam devel-

Table 4. Amalgam combinations that will cure hard at 100°C and are fluid at room temperature after amalgamation.
Curing criterion: hardness >90 durometer.

Ni:Cu weight ratio	Ni + Cu powder charge weight (grams)										
	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
0.111	Y	Y	Y	Y	Y	Y	Y	S			
0.143	N	Y	Y	Y	Y	Y	Y	S			
0.200		Y	Y	Y	Y	Y	Y	Y			
0.250	N	Y		Y	Y	Y	Y	Y	S		
0.273		N	Y	Y		Y					
0.333	N		Y	Y		Y					
0.429			N	Y	Y	Y			Y	Y	S

N: amalgam did not set up in >8,000 minutes
S: amalgam set up at durometer hardness >10 before being placed in 100°C oven
Y: fluid amalgam, curing time <8,000 minutes

Table 5. Set-up times at 35°C for amalgam formation studied (minutes). Durometer hardness <10.

Ni:Cu weight ratio	Ni + Cu powder charge weight (grams)							
	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6
0.111	115	60	50	25	X	X	X	
0.143		110	40	25	X	X	X	
0.200		110	45	35	18	X	X	X
0.250		180	120	50	35	25	X	X
0.273			180	77	50	30	X	X
0.333			190	90	60	50	15	X
0.429				200	95	60	20	X

X: Set hard in amalgamation capsule

Table 6. Curing times for amalgam formulations at 100°C (minutes).
Curing criterion: durometer hardness >90.

Ni:Cu weight ratio	Ni + Cu powder charge weight (grams)										
	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9
0.111	1,000	1,000	400	145	85	75	40	S			
0.143	>15,000	750	700	600	115	110	35	S			
0.200		5,000	850	300	150	110	65	20			
0.250	>15,000	7,200		1,200	675	200	120	80	S		
0.273		>12,000	1,300	1,210		1,200					
0.333	>15,000		1,550	1300	650						
0.429			12,500	7,000	1,050	700			90	85	S

S: amalgam set up at durometer D hardness >10, before placed in 100°C oven

Table 7. Properties of some Ga/Cu/Ni amalgams.

Properties	5% Ni- 20% Cu	5% Ni- 30% Cu	5% Ni- 40% Cu	10% Ni 20% Cu	10% Ni 40% Cu
Set-up time @ 35°C (minutes)	535	40	15	250	8
Curing time @ 100°C (hours)	90.7	12.1	1.8	68.1	1.1
Bond strength—alumina/alumina (kg/cm ²)	>16.5	>16.5	>16.5	>16.5	9.9
Bond strength Cu/Cu (kg)	>16.5	>16.5	>16.5	>16.5	12.7
Wetting—alumina	Good	Good	Good	Good	Good
Wetting—copper*	Fair	Good	Good	Good	Good
Bulk shear strength (kg/cm ²)	239.5	495.7	—	306.5	398.1
Ductility (percent elongation)	8.5	7.4	—	9.3	12.4
Electrical resistivity (mohm/cm)	18.2	11.8	10.75	—	12.1
Thermal expansion (μm/m°C)	—	21.7	—	—	—
Expansion on curing (percent)	30.7	11.6	20.6	15.1	8.4
Expansion on heating to 150°C (%)	1.36	0.06	0.07	0.95	-0.25
Thermal cycles, with zero failures	—	>3,000/-55 +150°C	—	—	—
Meniscus appearance	Smooth	Smooth	—	Smooth	Smooth

*Copper-coated alumina

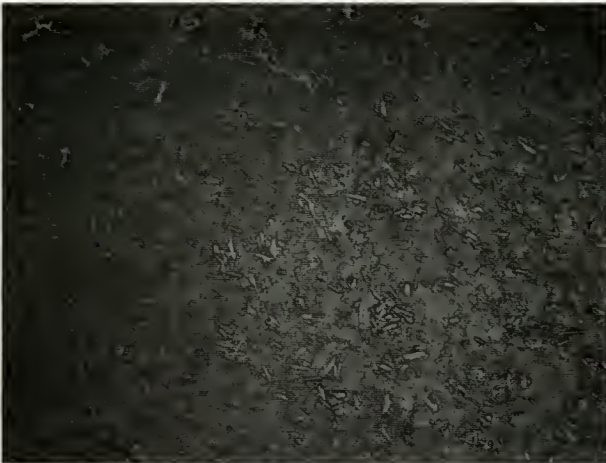


Figure 5. Microstructure of 25-percent silver/gallium alloy.

oped for bonding electronics. After an initial stage as a fully fluid mix of essentially dispersed powder particles that have a thin reaction layer of liquid-to-powder intermetallic, the material begins to stiffen. The reaction zone about each particle grows at the expense of the liquid, and free movement of the solid particles becomes constrained. This mechanism continues until all the particles lock together, and the material behaves as a solid. Ultimately, all the liquid chemically reacts, and the material becomes a matrix of reaction product containing remnant cores of the original powder elements.

**Table 8. Set-up times for Ga/Ag/Cu amalgam formulations at 35°C (minutes).
Durometer hardness <10.**

Ag:Cu weight ratio	Ag + Cu powder charge weight (grams)					
	1.1	1.2	1.3	1.4	1.5	1.6
0.111	143	85	45	X	X	
0.143	40	30	20	X	X	
0.200	45	45	15	X	X	X
0.250		250	250	20	10	X
0.273		75	30	20	X	X
0.333			120	45	20	X
0.429				45	25	20

X: Set hard in amalgamation capsule

Amalgamation experiments. Statistical experimental design matrix research indicates that amalgamation frequency, time, and composition are the most significant parameters in making a material that will remain fluid for several hours and yet harden at a few tens of degrees above room temperature. The time of amalgamation after the powder particles have been wetted is the most important part of a partitioned time variable. Powder particle size also affects results, with the finer powders reacting more quickly than coarser ones. Amalgamation involves two processes: 1) wetting of the powder

Table 9. Curing times for Ga/Ag/Cu amalgam formulations at 100°C (minutes).
Curing criterion: durometer hardness >90.

Ag:Cu weight ratio	Ag + Cu powder charge weight (grams)								
	0.9	1.0	1.1	1.2	1.3	1.4	1.5	1.6	1.7
0.111		1,300	800	805	680	180	80	60	40
0.143	2,900	2400	1,450	1,300	180	S			
0.200		1,600	1,550	1,575	1,200	1,300			
0.250		5,500	3,200	2,700	1,500	200	250	150	S
0.273		3,200	1,300	1,150	1,150	120	300	400	
0.333				2,000	4,500	3,700	275	1,000	
0.429				7,200	4,500	4,000	1,250	250	

S: amalgam set up at durometer hardness >10, before placed in 100°C oven

by the liquid component and 2) mechanical alloying arising through a hammering action in the presence of the liquid phase. During this second operation the thin rim of reaction product forms around each particle.

The shorter the alloying time after the powder particles have been wetted, the longer is the lifetime of the fluid amalgam at room temperature. In a commercial dental amalgamator where both wetting and mechanical alloying are performed at once, the two stages can be aurally separated. Mixing is silent until the particles are wetted, then the pestle moves freely in the fluid, making audible contact with the capsule body. A measurable time passes before the impact can be heard. It is shorter for higher vibration frequencies and longer for higher charge sizes in the capsule. The time depends upon surface conditions of the powder and the composition. As with set-up time, longer mechanical alloying times produce materials that are quicker to cure. Also, the higher the vibration frequency, the shorter will be the set-up and curing times. Too low a frequency, however, leads to unmixed or nonuniform materials, especially when the charge size is large.

Because the dental amalgamator performs both the wetting and mechanical alloying processes simultaneously, we cannot guarantee that every powder particle will be fully wetted before any particle begins to mechanically alloy. Consequently, these materials tend to show a range of set-up and curing times. Using a single overall time setting for mixing and amalgamating produced typical variations of about ± 18 minutes in the set-up time, with variations of ± 28 hours in room temperature curing times (normally 32 hours). Set-up times varied by ± 8 minutes when we employed a protocol

Table 10. Curing times for Ga/Ag/Ni amalgam formulations at 100°C (minutes). Curing criterion: durometer hardness >90.

Ag:Ni weight ratio	Ag + Ni powder charge weight (grams)							
	0.8	0.9	1.0	1.1	1.2	1.3	1.4	1.5
0.111	480	240	400	200	120	100	70	10
0.143	360	180	100	65	180	40	30	
0.200	120	120	75	100	75	60	45	
0.250	120	45	60	60	S			
0.273	180	180	120	300	120	120	10	
0.333				120	20			
0.429								

S: amalgam set up at durometer hardness >10, before placed in 100°C oven

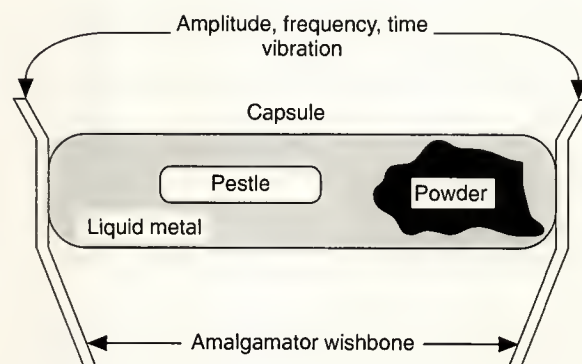


Figure 6. Diagram of dental amalgamator used to make amalgams.

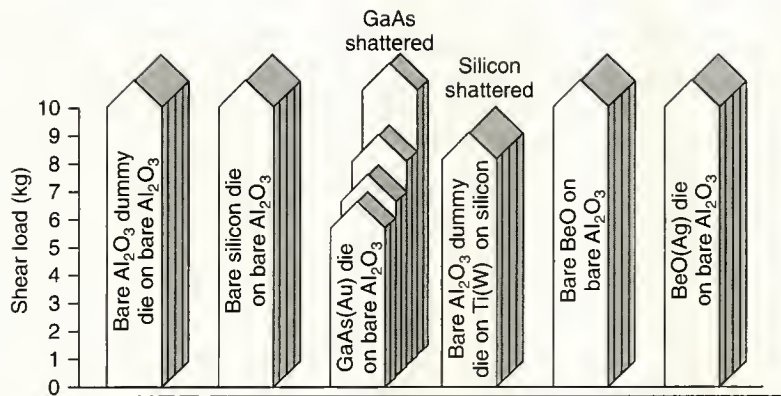


Figure 7. Shear strengths of some "dummy die" joints for some combinations of materials.

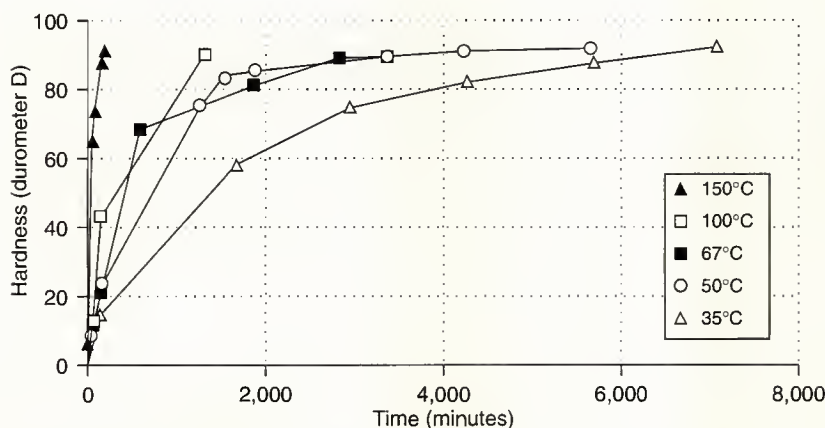


Figure 8. Hardening curves for a gallium/nickel/copper amalgam cured at a range of temperatures.

that used high frequency to wet the powder, stopped the mixing as soon as an aural signal was heard, reduced frequency, and then amalgamated for 5 seconds. When the powder was first wetted by hand, so that essentially no mechanical alloying was performed, and then placed in the amalgamator for 5 seconds to effect mechanical alloying, the variation fell to only ± 3 seconds. All samples cured to a hardness of 98 to 100 durometer C, which corresponds to between 200 Knoop and 460 Knoop.

Amalgam wetting. In the same way that wetting each powder particle requires the mechanical hammering with the amalgamator pestle, simple contact with the fluid amalgam cannot spontaneously wet surfaces. Some small amount of mechanical energy is required to cause the wetting. Wetting balance experiments using gallium indicate that temperatures must reach 170°C before gallium will spontaneously wet copper. We have used a simple jig in which a copiously wetted

copper slug is moved a fixed distance under the action of an approximately 20-gram load across the surface of the material to be wetted. The number of strokes required to completely wet the contacting area is called the scrub wetting number.

We have also experimented with various tin/lead alloy coatings and have investigated a range of liquid metal alloys and amalgams onto alumina and silicon surfaces, which were both bare and gold- and copper-coated. Experiments with alloyed gallium show that additions of nickel, antimony, and silver all significantly improve wetting on a wide range of substrates.

Amalgam joint strengths. We used assemblies having 0.25×0.375 -inch dummy die attached to 0.5×0.75 -inch substrates to investigate joint strengths between like and unlike materials. Initially, we tested the bonds with a Dage wirebond tester having a die shear attachment. Unfortunately, this setup permitted loads of up to only 10 kg, as this was the largest load cell available. Under these conditions most joints did not fail. This was true for tests on bare alumina and on copper- and gold-coated alumina for all curing times from 1.5 hours to 96 hours at 100°C for the binary gallium/nickel amalgam we used. Even when the "dummy die" size was reduced to 0.125×0.125 inches, strengths exceeded 10 kg. Figure 7 shows examples of joint strengths of some substrate-die combinations. Each bar in the diagram is a separate replicate and the " \wedge " top to each bar indicates that the bond did not fail.

Figure 8 shows one of a range of hardening curves for the ternary gallium/nickel/copper systems. From such curves we used the slopes of the initial reaction regions to determine Arrhenius plots from which we obtained the activation energies shown in Table 11. Activation energies in the range of 3 Kcal to 7 Kcal are consistent with a room temperature hardening mechanism.

Copper-to-copper bond strengths. Copper-to-copper bond strengths can be very high. See Figure 9, which shows strengths for joints tested in the die shear configuration. This figure also illustrates a slight problem we encountered, namely that of a wide spread in test values. The observed strengths ranged from about 15 kg to ~80 kg. For the die size tested (0.25×0.375 inches), die shear specifications require only 5-kg shear strength, so the observed strengths exceed requirements by from three to sixteen fold. Experiments at different

Table 11. Activation energies of Ga/Cu/Ni amalgams.

Ni:Cu Weight ratio	Activation energies (Kcals)				
	Ni + Cu powder charge weight (grams)				
	1.0	1.1	1.2	1.3	1.4
0.111	7.95				
0.143	3.58				
0.200	7.02				
0.250	6.85	5.10			
0.273		6.81	6.87		
0.333		5.65	6.05	6.18	
0.429			6.33	7.36	6.34

curing temperatures (50°C, 67°C, 100°C, and 150°C) showed very little dependence of strength upon curing temperature.

Application 1: Large area die attachment

To investigate the limits of die size that amalgam technology could bond, we prepared passive silicon die, metallized with different thicknesses of copper and nickel on chromium, in 0.500-, 0.700-, and 1.000-inch-square die. Also, we wanted to determine what type and thickness of metallization the process required when metallization was used. Ideally, if the metal thickness could be kept within that possible with a single, two-source sputter process, we could save one metallization step compared to the existing process. Consequently, we used copper thicknesses of 2,500Å and 5,000Å over the 700Å chromium adhesion layer, compared to the 0.5-mil copper and 0.5-mil nickel that had already shown success.¹ We hand-assembled individual die to alumina substrates with the same metallizations and fully cured them at 100°C. We then subjected each sample to a 10-kg shear acceptance test. We thermal-cycle tested the assembled die at -65°C to +150°C. Next, we shear tested a random sample of each combination to 10 kg after each 200 cycles up to 1,000 cycles. Thereafter we die shear tested all samples after 2,000 and 3,000 cycles of testing.

Thermal cycle performance. We considered thermal cycle performance an important materials screen, especially in the light of previous experience with the gallium/nickel system. Over the range of test conditions A, B, and C in MIL-STD-883, all gallium/nickel materials between 30-percent Ni and 45-percent Ni failed within at most 100 cycles by the

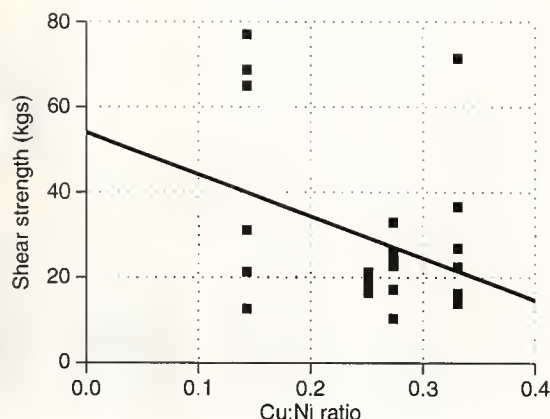


Figure 9. Shear strength of copper-to-copper joints applications trials.

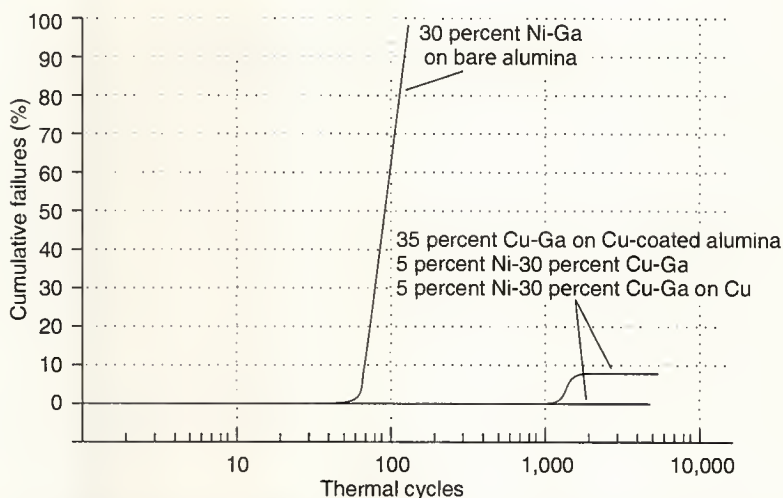


Figure 10. Cumulative failures v. thermal cycle performance of 0.5-inch-square silicon die on alumina.

swelling mechanism mentioned earlier. Figure 10 shows no failures for both 3,000 thermal cycles at -65°C to +150°C [MIL-STD-883C (Condition C)] and 1,200 thermal cycles plus 2,000 thermal shocks at -65°C to +150°C.

Doping the gallium with boron, tungsten, and aluminium, which has been shown to improve ductility of NiAl intermetallics by up to 48 percent,⁹ increased thermal cycle performance from 100 cycles at condition C to over 300 cycles. This was still insufficient, however, to hold promise of passing the 1,000-cycle acceptance criterion. Changing to the gallium/copper system produced samples that showed no failures. The gallium/copper/nickel formulation allowed 0.5-inch-square silicon-on-alumina (with copper metallization) to sur-

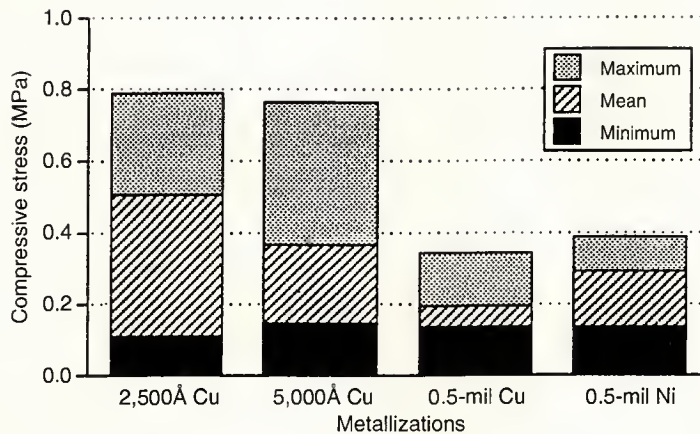


Figure 11. Die stress on 0.700-inch-square silicon die attached with amalgams.

vive both 3,000 condition C air-to-air thermal cycles and 1,200 air-to-air thermal cycles followed by 2,000 liquid-to-liquid condition C thermal shock cycles without any failures. Curing times also fell considerably.

A further study investigated both the metallization type and thickness requirements for bonding 0.7-inch-square and 1.0-inch-square silicon die to alumina. Provided a copper backing of between 10,000Å and 0.5 mil was used, we observed no thermal cycle failures even for the 1.0-inch-square die after 3,000 cycles.

Die bond stress. By measuring the curvature of the silicon die and knowing its thickness and the stress needed to produce that distortion, we showed that amalgam die attach-

ment produces die stresses of only 0.1 MPa to 0.8 MPa (see Figure 11). By comparison, solder die attachment produces stresses of about 15 MPa.

Application 2: Flip chips

The original motivation for investigating the use of amalgams for flip chips arose from requirements for face-down mounting of a color array detector. These silicon devices carry organic color filters deposited onto the surface making them particularly sensitive to both temperature and organic residues. This sensitivity precludes bonding with solders because of temperature and flux residues or with adhesives because of organic vapors and residues. Also, these devices were to be bonded to a glass substrate that was to act as the array window. As such, this study also served to demonstrate chip-on-glass capability. The device used had 3-mil-square peripheral contact pads, and the glass substrate had 3-mil-wide conductor lines of copper over a chromium adhesion layer.

One basic aim of using amalgams for flip-chip mounting was to have the solidification of a free-standing, dispensed-liquid "Hershey Kiss" provide the necessary stand-off height and to use the same amalgam as the bonding material. This combination could be achieved at will by dispensing on a chip-by-chip basis. To demonstrate the viability of amalgams for this type of bonding, we felt it necessary to first demonstrate the level of bonding these new materials could achieve. Then we needed to demonstrate that this strength was maintained when no bumps were used.

The first samples were plain aluminum-coated silicon die onto which we used an electroless nickel process to deposit small ~3-mil-square, 1-mil-high nickel bumps. We placed the bumps on the die to maintain a controlled gap between any organic material filters that might be placed upon the sensor devices that these chips simulated and the glass substrates that would act as sensor windows in the final application. When aligned, the bumps seated directly onto the copper conductors of the test substrates. This system led to problems such as extrusion of the dispensed amalgam from the bump-conductor track gap. The amalgam ran down the bumps and attacked the bare aluminum on the dummy die. Once we had demonstrated adhesion, we investigated the bumpless process. For this, we replaced the aluminum dummy die with a panel copper-plated glass slide. We applied the dispensing pattern consistent with the test chip to only one side of the component-substrate couple.

Amalgam dispensing. During this study, we developed the capability to accurately and repeatably dispense controlled amounts of amalgam in the form of Hershey Kiss bumps. Using the ultrasonics of the modified wirebonder shown in Figure 12 (as per US Patent 4,704,908), we applied the mechanical wetting motion by scrubbing the wetted capillary

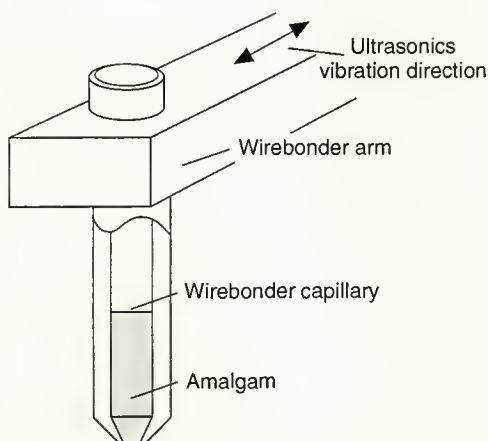


Figure 12. Diagram of modified wirebonder used to dispense 'Hershey Kisses' amalgam bumps.

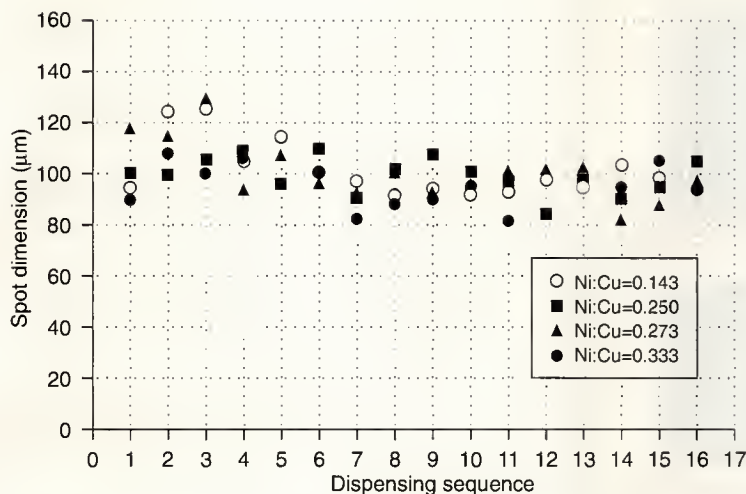


Figure 13. Dispensed dot diameters, 16 consecutive replicates.

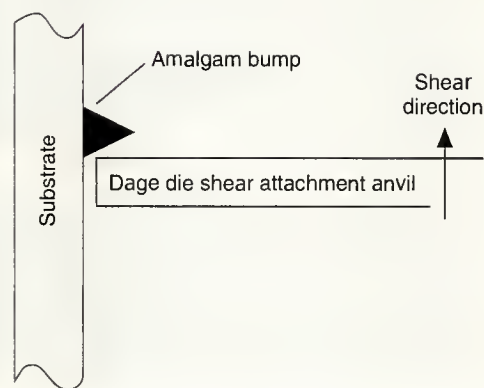


Figure 14. Amalgam bump die shear set-up.

tube tip onto the contact surface. When the capillary was raised, the surface tension of the wetted interface drew a thread of liquid from within the capillary to leave a Hershey Kiss deposit wetted to the substrate surface. By selecting a suitable capillary, we produced deposits only 0.003 inch at their base. Figure 13 indicates typical reproducibility of dot diameter for a range of amalgam compositions. With this process, dot height repeatability was similar to dot diameter repeatability. Using a micro die shear attachment on the Dage tester as shown in Figure 14, we measured adhesive strength of the dots to the substrate. As shown in Figure 15, shear strengths of the bumps ranged from 30 grams to 80 grams.

Flip-chip assembly. We assembled the completed flip-chip bonded assembly shown in Figure 16 with our adhesive bonder. We could control the temperature of the bonder base up to 120°C. We could also control its upper attachment arm by distance above a substrate or by contact pressure. In this instance we used a 1-mil gold wire, both to demonstrate that we had achieved the 1-mil stand-off and also as a back-up precaution. For flip-chip assembly we aligned the chip and substrate with amalgam dispensed on each. With positioning proximity controlled to maintain a standard stand-off height, we then lowered the device onto the substrate so that the amalgam bumps on each side melded, achieving the correct stand-off height. Next, we rapidly heated the base stage to between 100°C and 120°C and held it there for ~1 minute. The assembly then was stiff enough to be transferred to an air oven for curing at 67°C, a safe temperature for the organic filter materials of the proposed optic array.

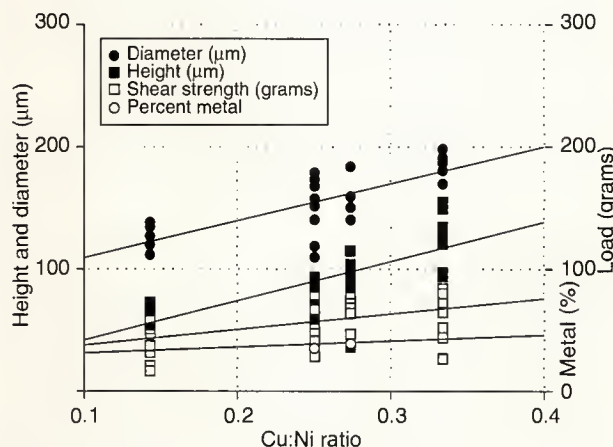


Figure 15. Amalgam bump dimensions and bond strength for a range of amalgam compositions.

THIS SUCCESSFUL ASSEMBLY DEMONSTRATES the feasibility of the amalgam approach, but continuity, functional, and environmental testing must still be performed. Also, within the purview of the original problem, we must still demonstrate hermetic sealing of an array of such devices. We have demonstrated the feasibility of using these materials for applications where conventional solders cannot be employed. We must still see if amalgams can more fully replace solders



Figure 16. Amalgam flip chip on glass, 0.003-inch-diameter amalgam bumps with 0.001-inch stand-off.

in applications such as surface-mount technology. On the basis of the results this article presents, we think that such a demonstration is feasible and is a logical step in the search for alternative, lead-free bonding materials for electronic and microelectronic applications. □

References

1. C.A. MacKay, "Amalgams for Alternative Bonding Materials," *Proc. Int'l. Electronics Packaging Soc. (IEPS) Conf.*, 1989, p. 1259.
2. G. Schuldt and C.A. MacKay, "Applications of Amalgams in Microelectronic Bonding," *Proc. Microelectronics Materials and Processing Conf.*, Am. Soc. of Metals (ASM) Int'l, 1992, pp. 141-147.
3. C.A. MacKay, "Bonding Metallurgies-Amalgams," Tech. Publication P/I 307-87, Microelectronics and Computer Technology Corp., Austin, Texas, 1987.
4. C.A. MacKay, "Parameters and Conditions for Making Usable Amalgams," Tech. Publication P/I 107-88, MCC, 1988.
5. C.A. MacKay, "Flip Chip Attach Using Amalgams Phase I: Viability Study," Tech. Publication P/I 332-91(Q), MCC, 1991.
6. C.A. MacKay, G. Schuldt, and Shih Hsu, "Large Area Die Attach: Materials Screening," Tech. Publication HVE XXX-92(Q), MCC, 1992.
7. G. Harmon, "Hard Gallium Alloys for Use as Low Contact Resistance Electrodes," *Rev. Soc. Insts.*, Vol. 31, No. 7, Jul. 1960, pp. 717-720.

8. I. Baker and P.R. Munroe, "Improving Intermetallic Ductility," *Jour. Minerals, Metals, & Materials Soc.*, Feb. 1988, pp. 28-31.
9. P.R. Munroe and I. Baker, "Structural Intermetallic Compounds," *Metals & Materials*, July 1988, pp. 435-438.
10. S. Takeuchi and E. Kuramoto, "Low Melting Point Ga-Cu Solders," *Acta Met.*, 1973, Vol. 21, No. 4, pp. 415-425.
11. H.A. Lipsitt, D. Shechtman, and R.E. Schafrik, "Deformation and Fracture of Ti_3Al at Elevated Temperatures," *Metallurgical Trans. A*, Vol. 11A, 1979, pp. 1369-1375.
12. C.T. Lui, C.L. White, and J.A. Horton, "Effect of Boron on Grain Boundaries in Ni_3Al ," *Acta Met.*, Vol. 33, No. 2, 1985, pp. 312-229.



Colin A. MacKay is the principal member of the technical staff at the Microelectronics and Computer Technology Corporation, where he is involved with its ongoing bonding programs and leads the amalgams development project. His interests include a wide range of bonding and metallurgical subjects.

MacKay earned his BSc, MSc, and PhD at London University. He has published more than 70 technical papers on a broad range of metallurgical and joining subjects. Currently, he is a member of the Institute of Printed Circuits, the Institute of Physics (Britain), the Institute of Metallurgists, the Institute of Metal Finishing, the British Association of Brazing and Soldering, the International Society of Hybrid Microelectronics (ISAHM), and the Institute of Electrical and Electronics Engineers. He presently chairs the Electronic Materials Division of the American Society of Metals.

Direct questions about this article to the author at the Microelectronics and Computer Technology Corporation, 12100 Technology Boulevard, Austin, Texas 78727; via fax at (512) 250-2893; or via e-mail at mackay@mcc.com.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 165

Medium 166

High 167



A 16-Kbit Θ -Search Associative Memory

Advances in VLSI technology combined with recent applications proposed for associative processing have given new impetus to the design of cost-effective associative memories. We introduce the design of a high-performance, high-capacity Θ -search associative memory ($\Theta \in \{<, >, \leq, \geq, =, \neq\}$). PSPICE simulations and layouts show that the proposed Θ -search associative chip consisting of 256 words, each 64-bits long, can fit on a $13.5 \times 9.5\text{-mm}^2$ chip. Further, it can perform maskable Θ -search operations over its contents in 110 ns.

A. R. Hurson

Patrick M. Miller

Pennsylvania State
University

The concept of associative memory and its applications have been the subject of much research since the mid-1960s.¹⁻³ Associative memories have an "active nature"—they perform operations at the memory level. This active nature, combined with the embedded parallelism inherent in associative operations, has made associative memories a suitable environment for applications that require massive simple arithmetic or search operations. Despite these benefits, however, other drawbacks have worked against widespread use of this technology. Compared to random access storage bits, associative storage is more complex. This complexity leads to higher costs and reduced capacity, penalties that undermine the practicality and feasibility of associative memory.

At first glance, this complexity seems to be a valid concern. After all, compared to a conventional random access memory, a fully parallel associative memory suffers the additional cost of search circuitry at every bit position. Such devices also require more complex control circuitry to operate them. In some cases, though, this size and cost increase should be acceptable. For example, the dynamic equality search-bit cell proposed by Storman et al.⁴ is only 1.5 times the size of the fully static CMOS RAM cell described in Iizuka et al.⁵— $1,656 \lambda^2$ versus $1,122 \lambda^2$. The increased functionality and parallelism of the associative memory more than offsets this area penalty.

Recently, researchers have turned to the design and fabrication of associative memories with increasing interest.^{4,6-16} This surge of curiosity is mainly due to

- advances in technology that allow the fabrication of high-yield, high-density, complex chips and
- new applications that have found associative operations very appealing to their performances.^{4,6,8,13,14,17}

The so-called Θ -search associative memories are a class of associative chips that can be used in database applications.^{6,17} In such an environment one can perform Θ -search operations effectively in hardware (where Θ is a relational operator— $\Theta \in \{<, >, \leq, \geq, =, \neq\}$).

Θ -search parallel associative memories

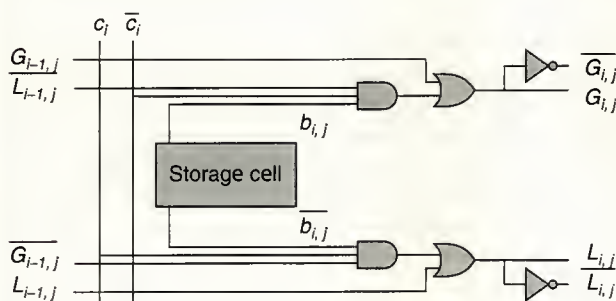
The parallel search ability of the associative memories makes them well suited to performing database operations directly at the memory level. In particular, we have seen that an associative memory with maskable Θ -search capability is sufficient to effectively implement the set of relational operators.¹⁷ Other associative memory designs^{9,12,15} perform Θ searches directly in hardware, but they differ in the speed and size of the associative cell.

The maskable parallel Θ -search associative memory addressed by Hurson and Shirazi⁹ determines the Θ relations in a cascade fashion from the most significant bit to the least significant bit. To simplify the memory cell:

- control signals (such as word select signal and search signal) are used at the word level rather than the bit level, and

- Θ searches are performed based on searches on less than and greater than.

Figure 1 depicts the general organization of an associative cell. In this figure the C_i and \overline{C}_i signals represent the content of the i th bit of comparand register (Comp) after it is masked (M). That is,

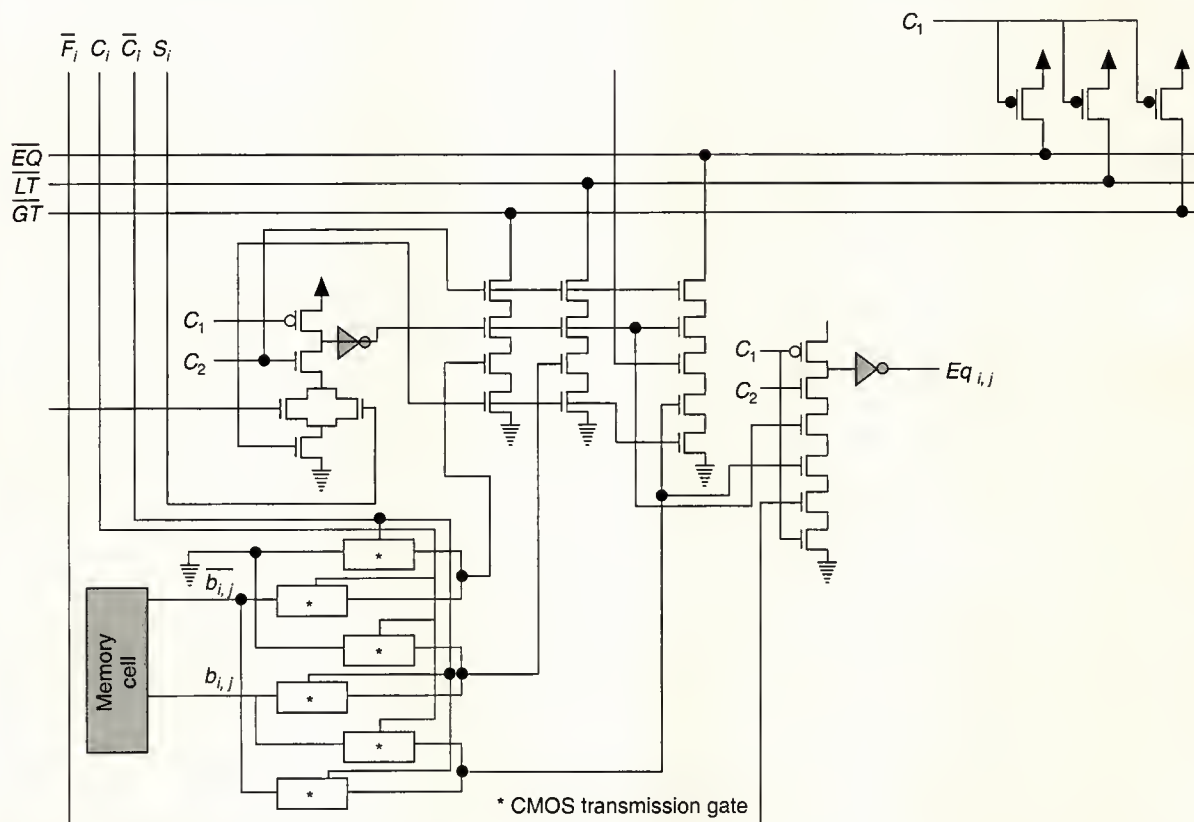
Figure 1. \ominus -search associative cell.⁹

$$C_i = 1 \quad \text{if } M_i = 1 \text{ and } \text{Comp}_i = 1$$

$$\overline{\mathcal{C}}_i = 1 \text{ if } M_i = 1 \text{ and } \text{Comp}_i = 0$$

For $1 \leq i \leq m$, where m is the word length, b_{ij} ($\overline{b_{ij}}$) represents the i th bit of the j th memory word. Each associative bit is a dynamic cell refreshing itself during the second phase of a two-phase clock. $L_{i-1,j}$ ($\overline{L_{i-1,j}}$) and $G_{i-1,j}$ ($\overline{G_{i-1,j}}$) are the greater-than and less-than signals generated from the left neighboring cell. L_{ij} ($\overline{L_{ij}}$) and G_{ij} ($\overline{G_{ij}}$) are similar signals generated by the i th bit of word j . The design includes a multiple maskable write and read operation. The cell size of the proposed design based on the NMOS technology was an estimated $5,000 \lambda^2$ with the search time proportional to the word length.

To minimize the search time, Petrie and Hurson¹⁵ introduced a Θ -search associative design based on a domino CMOS concept. This model used a precharged bus architecture to reduce the search time proportional to the attribute length rather than the word length. Figure 2 shows the organization of this design. In this figure \overline{EQ} , \overline{LT} , and \overline{GT} are precharged buses, C_1 and C_2 are clock pulses, and Eq (equality-so-far) is a signal generated

Figure 2. Domino Θ -search associative cell.¹⁵

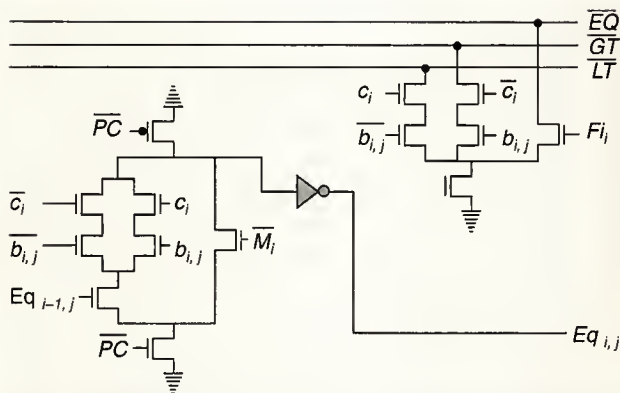


Figure 3. O(A)Θ-search associative cell.¹²

in each cell and passed to its right neighbor. Finally, S_i and F_i signals realized by the equations $\bar{M}_{i-1}M_i$ and $M_{i-1}\bar{M}_i$, respectively, represent the beginning and end positions of the search field. Taking routing into consideration, the search circuitry cell size of this design is an estimated $13,056 \lambda^2$.

Of the two Θ-search associative cell designs mentioned earlier, Petrie and Hurson's model is the most attractive since its search time is proportional to the attribute length. However, an associative cell using this search circuitry contains over 7.5 times the number of transistors found in a static RAM cell. Despite its increased performance, this is too high a price to pay for the associative memory. For this reason, Miller et al.¹² attempted to minimize the size of this model. The resulting design uses precharged buses to achieve the search time proportional to the attribute length. However, the new cell requires only 16 transistors per bit for the search circuitry, instead of 44 transistors per bit found in the original design (see Figure 3). In this design a signal is propagated from one cell to its less significant neighbor via domino CMOS design (for example, Eq_i). F_i , a finish signal indicating the last position of the search field, is generated by the equation $F_i = M_{i-1}\bar{M}_i$. The size of each associative cell is estimated as $6,528 \lambda^2$. While this is smaller than Petrie and Hurson's design, it is still too large to be practical in an associative memory.

Proposed Θ-search associative cell

Despite the advantages of associative processing, there are very few associative memories available in the marketplace either as general-purpose chips or as components in standard cell libraries for VLSI design. As discussed earlier, the perceived high cost of associative memory may contribute to this shortage. However, such a perception might not be the only reason that has discouraged the mass production of associative memories. As Miller et al. discuss, applications that use associative operations often require associative memories that perform different low-level operations. For example,

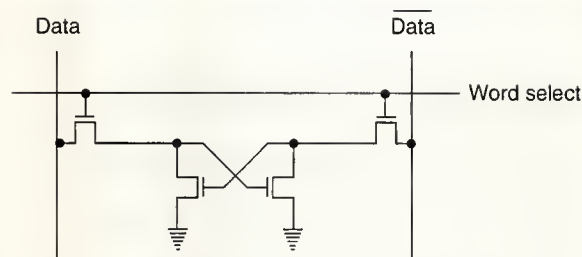


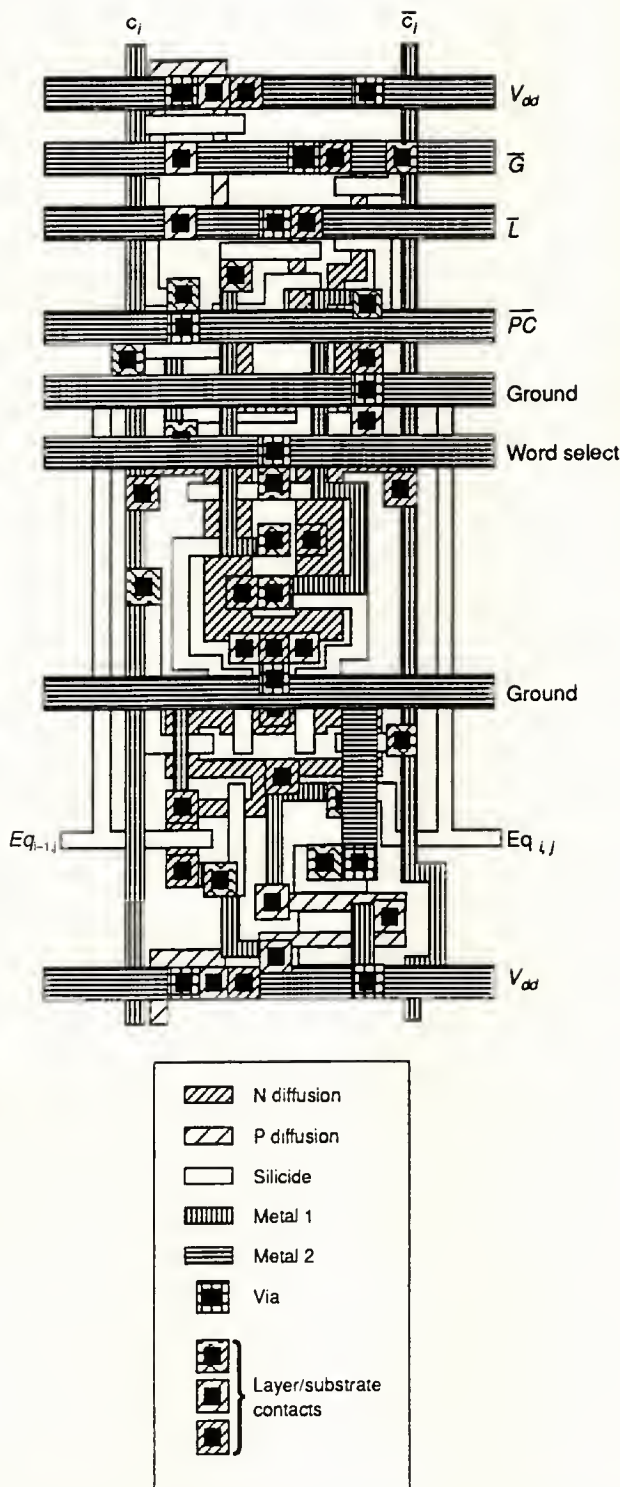
Figure 4. Four-transistor dynamic memory element.

some require maskable write operations and some do not. Therefore, we must develop a scheme that allows easy implementation and fabrication of associative memories for supporting the needed operations efficiently. This scheme requires a design methodology with a high degree of modularity, independence, and compatibility among various elements of an associative memory. The design we propose is a step in this direction while offering a better area use than found in other Θ-search associative cells.

Memory cell design. The purpose of the memory element is to store a single bit of information in an associative word. Originally, we considered static CMOS elements because of their design simplicity and their ability to hold data without the need for refreshing. Our investigation showed that such an approach requires various designs for supporting different memory operation features. Furthermore, the cell size was too large to practically support a maskable multiple-write operation.

As a next step, we examined one-transistor dynamic memory. While this design offered a better size (a third of the static memory), it required a large refresh cycle, one that was proportional to the number of associative words. Further investigation revealed that a four-transistor dynamic element design was the best choice for the memory cell.¹⁸

Shown in Figure 4 is a four-transistor dynamic memory element.¹⁰ The cell is basically a cross-coupled, pull-down pair of transistors. During a normal write operation, the data buses (C_i and \bar{C}_i) are set to opposite values, and the word select line is driven high. To read, both data buses are charged high and the pass transistors are turned on. Note that the read operation is not destructive. All memory words are refreshed simultaneously by holding both data buses high and opening the pass transistors. In addition, the design of the cell is independent of the overall functionality of the associative module. The cell size is an estimated $37 \lambda \times 28 \lambda$ —one third the size of its static counterpart. It is slightly smaller than the three-transistor, multiple-maskable memory element evolved from the one-transistor memory element mentioned earlier. The four-transistor cell has a tighter layout and does not need a MOS capacitor. Note that we used Pucknell and


 Figure 5. Layout of the proposed Θ -search associative cell.

Eshraghian's design rules¹⁹ for all layout estimates. Such a set of design rules is based on a trilayer-metal CMOS technology.

We ran PSPICE simulations to determine the read, write, and refresh time of the memory cell. Results showed that when the parasitic capacitances are considered, the write and refresh times are 6 ns and 10 ns, respectively. Moreover, after data buses ($7,296\lambda$ long) are established at their proper voltages, the read time is approximately 10 ns. The simulations used SPICE parameters for a typical CMOS P-well process. Miller¹⁸ provides a more detailed discussion regarding the simulation results.

Θ -search circuitry design. The main reason for the large size of Petrie and Hurson's Θ -search circuitry is the number of vertical and horizontal buses passing through each cell. Therefore, we directed our efforts toward reducing the number of these buses. First, as Hurson and Shirazi report, different variations of Θ -search operation can be performed by greater-than and less-than comparisons (See Figure 1). As a result, one can eliminate the E_q bus line from the designs introduced in Miller et al.¹² and Petrie and Hurson.¹⁵ Second, the \bar{M}_i vertical bus and its associated transistor can be removed from Miller's proposed design (Figure 3), since the masking information can be easily determined from the C_i and \bar{C}_i buses. (They are both 0 when the bit is masked out). Third, our analysis revealed that the \bar{PC} vertical bus can be replaced by a horizontal \bar{PC} bus that reduces the cell size. Finally, a more compact design of the E_q generator (Figures 2 and 3) will reduce the cell size.

Figure 5 shows the layout of the proposed Θ -search circuitry. This layout can be divided into three parts. The upper third contains the G_j and L_j search buses and the associated pull-down logic (Figure 6). Note that there is enough room to include precharge transistors in the associative cell design. The middle part of the layout contains the four-transistor dynamic memory element (Figure 4). The bottom third of the layout generates the $E_{q,i,j}$ signal needed by the next associative cell ($i+1, j$)—see Figure 7.

The total size of the proposed design is about $3,652\lambda^2$. This is less than half of Petrie and Hurson's design and a little bit over half the size of Miller's design. We ran PSPICE simulations to determine the timing characteristics of the proposed design. In the worst case (equality search) the search time is $(1.65m + 4)$ ns after parasitic capacitances are considered. For $m = 64$, the search time is less than 110 ns. Table 1 compares the characteristics of the proposed Θ -search circuitry against competing models^{9,12,15} for a chip memory area of $11.6 \times 7.3 \text{ mm}^2$. Miller¹⁸ shows that if the control and supporting circuitry were added to the proposed design, a 16-Kbit associative chip could fit in $13.5 \times 9.5 \text{ mm}^2$.

Extending the physical word length

A fixed-length associative memory does not lend itself to handling variable record structures commonly used in artificial intelligence and database applications. Therefore, to in-

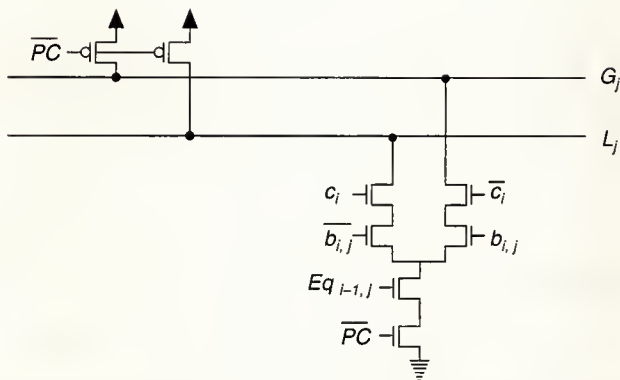


Figure 6. G_j and L_j pull-down logic.

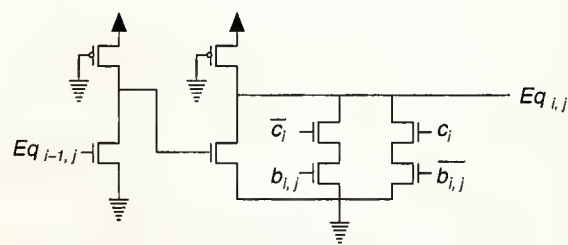


Figure 7. Θ -search Eq generator logic.

crease the versatility of associative memory in handling variable and long word lengths, one has to devise proper schemes to increase physical length of associative words. Two general approaches have been proposed.

The first approach allows K associative chips of N m -bit words to be linked together horizontally to make a memory of size $N \times Km$ words.¹⁷ In this organization each m -bit segment of a long-word (Lword) is stored at the same address across the K associative chips. Lwords can be searched in one cycle. To determine the proper Lwords that satisfy the search conditions, each chip outputs the address of its first matched word. An outside controller examines these addresses. If they all match, the words stored at that address comprise the desired Lword. Otherwise, the controller issues a command to the chip with the lowest match address to output the address of the next matched word. The addresses are compared again, and this process continues until matched Lwords are found or one of the chips has no more matched words. Once the locations of all the matching Lwords have been determined, each Lword can be read through K consecutive reads of the associative chips. This scheme allows the Lword search to be performed in one cycle, but locating all the matching Lwords may take as many as N additional

Table 1. Comparative analysis of Θ -search associative memories.

Model	Size including memory element (λ^2)	Θ -search time (ns)	Capacity (Kbits)
Hurson & Shirazi ⁹	5,000	139	16.0
*Petrie & Hurson ¹⁵	14,092	67	5.7
*Miller, Hurson			
Hettmansperger ¹²	9,544	40	8.4
Proposed scheme	5,016	110	16.0

Word length = 64 bits

* Attribute length = 16 bits

* Chip memory area = $11.6 \times 7.3 \text{ mm}^2$

cycles (that is, the number of words in each associative chip).

The second approach stores $(K \times m)$ -bit Lword as K consecutive m -bit words in an N -word associative chip. This approach requires K search cycles to search an entire Lword. If proper logic is included in the tag flags, all matching Lwords can be identified immediately. Since in most practical applications K is smaller than N , this approach is faster than the former scheme. Two mechanisms of delimiting Lwords in this scheme have been addressed: identifying the first segment of each Lword via a tag bit attached to the words,⁴ or assigning a sequence number to each m -segments of an Lword.¹⁴ Both mechanisms provide an Lword search mode where a tag bit is not set unless the current search criteria is met and the previous word's tag flag is set.

Searching an Lword proceeds as follows. First, the initial segment of proper Lwords is marked by a normal search, and thereafter each segment is searched in sequence using the Lword-search mode. At the end of this process the addresses of the Lwords that have satisfied the search conditions need to be determined. To determine which Lwords matched the search criteria, one proposed scheme⁴ has the execution time proportional to the number of Lwords satisfying the search condition. A second proposed scheme¹⁴ requires an execution time complexity of K .

We propose a new scheme built upon the second approach that marks all segments of matching Lwords without any timing penalty but at the expense of a small hardware overhead. The general idea is that, at the end of a series of search operations, all the segments of an Lword are marked if any one of the segments' tag flags are set. (If the Lword does not match the search criteria, none of its segments' tag flags will be set.) To do this, first, a first-segment flag (F_s) is added to show where a new Lword starts. That is,

$$F_{sj} = \begin{cases} 1 & \text{if word } j \text{ is the first segment of an Lword} \\ 0 & \text{otherwise} \end{cases}$$

Second, each associative word (j) generates two signals, namely $fmatch_j$ and $bmatch_j$. The $fmatch_j$ signal is set to 1 if the current or any of the previous segments in an Lword has a tag flag set. Also, $bmatch_j$ is 1 if the current or any of the following segments of an Lword has its tag set. These conditions are given by:

$$fmatch_j = fmatch_{j-1} \overline{F_{sj}} + Tag_j \text{ and} \\ bmatch_j = F_{sj} (bmatch_{j-1} + Tag_j)$$

The Lword search operation is performed as a sequence of the following steps:

1. Search **Lfirst** to set the tag bit of first segment of all potential Lwords: $Tag_j = F_{sj} Tag_j$
2. Perform a sequence of (at most $K-1$) **Lsearch** operations: $Tag_j = \overline{F_{sj}} Tag_{j-1} + Tag_j$
3. Generate and propagate the $fmatch_j$ and $bmatch_j$ signals
4. Perform a **collection operation** in parallel over all the words: $Tag_j = fmatch_j + bmatch_{j+1}$

This sequence causes the tags of all segments in a matching Lword to be set. Note that step 3 will be done concurrently with step 2. The only requirement is that the $fmatch$ and $bmatch$ signals be stable before the collection operation is initiated. The setting time of these signals is proportional to K (that is, $K\mathfrak{T}$ where \mathfrak{T} is the propagation delay of the $fmatch/bmatch$ generation logic). Simulations estimate that $\mathfrak{T} = 1.25$ ns for a $\lambda = 1 \mu\text{m}$ CMOS process, so a total time of $1.25 K$ ns must separate the last Lsearch operation and the initiation of the collection operation.

Unless associative operations are issued back to back and K is large, these signals will be stable before the collection operation occurs. For large values of K , several collection operations could be initiated in a row to ensure the proper set-up time for the $fmatch/bmatch$ logic. In any case, our scheme is still faster than the scheme proposed in Stormon et al.⁴ An interesting consequence of our method is that Lword searches no longer need to start at the first segment of the Lword. This feature is accomplished by storing a segment number in each associative word and performing an equality search over the segment field based on the starting segment number.

FUTURE ASSOCIATIVE MEMORY RESEARCH lies in two directions. To develop modular associative chips, work will continue to optimize the size of various components of an associative chip. These components will be defined to allow

the fabrication of special-purpose chips. A user-friendly, interactive, and intelligent software package should also be developed to assist the user in laying out an associative chip with proper functionality. In our continuing work we have made progress in using so-called genetic algorithms as an optimizing tool to develop customized associative chips. \square

References

1. C.C. Foster, *Content Addressable Parallel Processors*, Van Nostrand Reinhold, New York, 1976.
2. T. Kohonen, *Content Addressable Memories*, Springer-Verlag, New York, 1980.
3. S.S. Yau and H.S. Fung, "Associative Processor Architecture—A Survey," *ACM Computing Systems*, Vol. 9, No. 1, 1977, pp. 3-27.
4. C.D. Stormon et al., "An Architecture Based on Content-Addressable Memory for Rapid Execution of Prolog," *Proc. 5th Int'l Conf. Logic Programming*, 1988, pp. 1448-1473.
5. T. Iizuka et al., "Fully Static 16-Kb Bulk CMOS RAM," *Digest of Technical Papers, IEEE Int'l Solid-State Conf.*, 1980, pp. 226-227.
6. J.-B. Cheng and A.R. Hurson "A Hardware Approach to the Parallel Finite-State Automaton for Text Retrieval Operations," *J. Computer-Aided VLSI Design*, Vol. 2, No. 1, 1990, pp. 65-81.
7. K.E. Grosspietsch, "Associative Processors and Memories: A Survey," *IEEE Micro*, Vol. 12, No. 3, June 1992, pp. 12-19.
8. F. Herrmann and C. Sodini, "A Dynamic Associative Processor for Machine Vision Applications," *IEEE Micro*, Vol. 12, No. 3, June 1992, pp. 31-41.
9. A.R. Hurson and B. Shirazi, "Associative Memories: Has Their Time Come? Applications and VLSI Complexities," *Proc. 20th Int'l Conf. Systems Sciences*, IEEE Computer Society Press, Los Alamitos, Calif., 1987, pp. 284-292.
10. S.R. Jones et al., "A 9-Kbit Associative Memory for High-Speed Parallel Processing," *IEEE J. Solid-State Circuits*, Vol. 23, No. 2, 1988, pp. 543-548.
11. H. Kadota et al., "An 8-Kbit Content-Addressable and Reentrant Memory," *IEEE J. Solid-State Circuits*, Vol. SC-20, No. 5, 1985, pp. 951-957.
12. P.M. Miller, A.R. Hurson, and R.H. Hettmansperger, "A Modular Scheme for Designing Special Purpose Associative Memories," in press *J. Computer Aided VLSI Design*.
13. J. Naganuma et al., "High Speed CAM Based Architecture for a Prolog Machine (ASCA)," *IEEE Trans. Computers*, CS Press, Los Alamitos, Calif., Vol. C-37, No. 11, 1988, pp. 1375-1383.
14. T. Ogura et al., "A 20-Kb CMOS Associative Memory LSI for Artificial Intelligence Machines," *IEEE J. Solid State Circuits*, Vol. 24, No. 8, 1989, pp. 1014-1020.
15. C.R. Petrie and A.R. Hurson, "A VLSI Join Module," *VLSI Systems Design*, Vol. 9, No. 10, 1988, pp. 46-58.
16. J. Wade and C. Sodini, "A Ternary Content Addressable Search



April 1993 issue (card void after October 1993)

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone(_____) _____

Reader Interest

(Add comments on the back)

Readers,
Indicate your interest in
articles and departments
by **circling the appropriate number** (shown on
the last page of articles
and departments):

150 151 152 180 181 182
153 154 155 183 184 185
156 157 158 186 187 188

159 160 161 189 190 191
162 163 164 192 193 194
165 166 167 195 196 197

168 169 170 198 199 200
171 172 173 201 202 203
174 175 176 204 205 206

177 178 179 207 208 209

Product Information

(Circle the numbers to receive
product information)

1	21	41	61	81	101	121	141
2	22	42	62	82	102	122	142
3	23	43	63	83	103	123	143
4	24	44	64	84	104	124	144
5	25	45	65	85	105	125	145
6	26	46	66	86	106	126	146
7	27	47	67	87	107	127	147
8	28	48	68	88	108	128	148
9	29	49	69	89	109	129	149
10	30	50	70	90	110	130	
11	31	51	71	91	111	131	
12	32	52	72	92	112	132	
13	33	53	73	93	113	133	
14	34	54	74	94	114	134	
15	35	55	75	95	115	135	
16	36	56	76	96	116	136	
17	37	57	77	97	117	137	
18	38	58	78	98	118	138	
19	39	59	79	99	119	139	
20	40	60	80	100	120	140	



April 1993 issue (card void after October 1993)

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone(_____) _____

Reader Interest

(Add comments on the back)

Readers,
Indicate your interest in
articles and departments
by **circling the appropriate number** (shown on
the last page of articles
and departments):

150 151 152 180 181 182
153 154 155 183 184 185
156 157 158 186 187 188

159 160 161 189 190 191
162 163 164 192 193 194
165 166 167 195 196 197

168 169 170 198 199 200
171 172 173 201 202 203
174 175 176 204 205 206

177 178 179 207 208 209

Product Information

(Circle the numbers to receive
product information)

1	21	41	61	81	101	121	141
2	22	42	62	82	102	122	142
3	23	43	63	83	103	123	143
4	24	44	64	84	104	124	144
5	25	45	65	85	105	125	145
6	26	46	66	86	106	126	146
7	27	47	67	87	107	127	147
8	28	48	68	88	108	128	148
9	29	49	69	89	109	129	149
10	30	50	70	90	110	130	
11	31	51	71	91	111	131	
12	32	52	72	92	112	132	
13	33	53	73	93	113	133	
14	34	54	74	94	114	134	
15	35	55	75	95	115	135	
16	36	56	76	96	116	136	
17	37	57	77	97	117	137	
18	38	58	78	98	118	138	
19	39	59	79	99	119	139	
20	40	60	80	100	120	140	

SUBSCRIBE TO IEEE MICRO

All the facts about today's chips and systems

☐ **YES**, sign me up!

If you are a member of the Computer Society or any other IEEE society,
pay the member rate of only \$23 for a year's subscription (six issues).

Society: _____

IEEE membership no: _____

Society members: Subscriptions are annualized. For orders submitted March through
August, pay half the full-year rate (\$11.50) for three bimonthly issues.

Full Signature _____ Date _____

Name _____

Street _____

City _____

State/Country _____ ZIP/Postal Code _____

☐ **YES**, sign me up!

If you are a member of ACM, ACS, BCS, IEE (UK), IEEE but not a
member of an IEEE society), IECEJ, IPSJ, NSPE, SCS, or other
professional society, pay the sister-society rate of only \$42 for a year's
subscription (six issues).

Organization: _____ Membership no: _____

For information on airmail option, send fax request to (714) 821-4010.

☐ Payment enclosed *Residents of CA, DC, Canada, and Belgium add applicable sales tax.*

☐ Charge to ☐ Visa ☐ MasterCard ☐ American Express

Charge-card number _____

Expiration date _____

Month _____ Year _____

Prices valid through 12/31/93
493 MICRO

Charge orders also taken by phone:
(714) 821-8380 8 a.m. to 5 p.m. Pacific time
Circulation Dept.
10662 Los Vaqueros Circ., PO Box 3014
Los Alamitos, CA 90720-1264

Editorial comments

I liked: _____

I disliked: _____

I would like to see: _____

Reviewers Needed. If interested, send professional data to Dante Del Corso, Dipartimento di Elettronica, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129, Torino, Italy.

For reader service inquiries, see other side.



PLACE
POSTAGE
HERE

PO Box is for reader service cards only.

IEEE Micro

PO BOX 16508
NORTH HOLLYWOOD CA 91615-6508
USA



Editorial comments

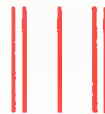
I liked: _____

I disliked: _____

I would like to see: _____

Reviewers Needed. If interested, send professional data to Dante Del Corso, Dipartimento di Elettronica, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129, Torino, Italy.

For reader service inquiries, see other side.

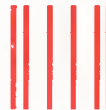


PLACE
POSTAGE
HERE

PO Box is for reader service cards only.

IEEE Micro

PO BOX 16508
NORTH HOLLYWOOD CA 91615-6508
USA



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 38 LOS ALAMITOS, CA

POSTAGE WILL BE PAID BY ADDRESSEE

IEEE COMPUTER SOCIETY

PO BOX 3014
LOS ALAMITOS CA 90720-9804
USA



Engine," *IEEE J. Solid State Circuits*, Vol. 24, No. 4, 1989, pp. 1003-1013.

17. A.R. Hurson et al., "A Reconfigurable MIMD Backend Database Machine," *J. Parallel and Distributed Computing*, Vol. 11, 1991, pp. 37-50.
18. P.M. Miller, *A Modular Organization for Designing Special-Purpose Associative Memories*, master's thesis, Dept. of Electrical and Computer Engineering, Pennsylvania State University, University Park, Pa., 1991.
19. D.A. Pucknell and K. Eshraghian, *Basic VLSI Design: Systems and Circuits*, Prentice Hall, New York, 1988.



A.R. Hurson is on the computer engineering faculty at Pennsylvania State University. For the past 12 years, his research has been directed toward the design and analysis of general- and special-purpose computer architectures.

Hurson has published more than 110 technical papers. He is the coauthor of *IEEE Tutorial on Parallel Architectures for Database Systems* and the cofounder of the IEEE Symposium on Parallel and Distributed Processing. He is a member of the Computer Society Press editorial board and the IEEE Distinguished Visitors Program.



Patrick M. Miller is a software engineer at AT&T Bell Laboratories where his interests include query performance, databases, object-oriented design, and process management.

He received his BS and MS from Pennsylvania State University, where his work concentrated on concurrent architectures, associative memory, and VLSI design.

Direct any comments regarding this article to A.R. Hurson, Department of Electrical and Computer Engineering, Pennsylvania State University, University Park, PA 16802 or e-mail at a2h@ecl.psu.edu.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 168

Medium 169

High 170

IEEE COMPUTER SOCIETY PRESS

VISUALIZATION '92 — 3rd IEEE Conference on Visualization

— OCTOBER 19-21, 1992 — BOSTON, MA —

Sections: Volume and Flow Visualization, Surfaces Visualization, Visualization Systems, Data Structures, Visual Programming, Visualization Software Systems and Techniques, Virtual Reality and Multimedia, Visualization Methods, Terrain Visualization, Case Studies in Physics, Life Sciences, Environments and Fluid Dynamics.

504 PAGES. SOFTCOVER. ISBN 0-8186-2897-9.
CATALOG # 2897-02 — \$120.00 MEMBERS \$60.00

PHYSCOMP '92 — Workshop on Physics and Computation

— OCTOBER 2-4, 1992 — DALLAS, TX —

Articles: Information is Physical, Artificial Physics, Physical Laws and Information Content, Computational Complexity and Phase Transitions, The Energy Content of Knowledge, Physical Information Theory, Quantum Mechanical Neural Networks, Physical Meaning of Computation, Oracle Quantum Computing, Information in Dynamics, Physics Analogs in Communication Models, Aspects of Computability in Physics, and more.

344 PAGES. SOFTCOVER. ISBN 0-8186-3420-0.
CATALOG # 3420-02 — \$70.00 MEMBERS \$35.00

CVPR '92 — IEEE Computer Society Conference on Computer Vision and Pattern Recognition

— JUNE 15-18, 1992 — CHAMPAIGN, IL —

Sections: Shape Description and Recovery, Calibration and Pose Estimation, Neural Networks, Motion Sequences and Optical Flow, Perceptual Organization and Curve Description, Shape from Texture / Focus, Object Recognition, Range Images, Applications, Navigation and Recognition, Shape from Shading / Photometric Stereo, Statistical Models for Low-Level Vision, Segmentation.

888 PAGES. SOFTCOVER. ISBN 0-8186-2855-3.
CATALOG # 2855-02 — \$130.00 MEMBERS \$65.00

— AVAILABLE AFTER 6/17/93 —

CVPR '93 —

— JUNE 15-17, 1993 — NYC, NY —

c.800 PAGES. SOFTCOVER. ISBN 0-8186-3880-X.
CATALOG # 3880-02 — *\$120.00 MEMBERS \$60.00
(* prepublication price)



To order call: 1-800-CS-BOOKS
or FAX 714-821-4010



Special Report: **Virtual Reality in Japan**

David Kahaner

US Office of Naval
Research

Using computer graphics and three-dimensional images, virtual reality makes things that do not actually exist appear to the operator as though they do. Unlike conventional two-dimensional techniques, this emerging technology gives users a "feel" for the objects before them. Current Japanese research seeks to use virtual reality for such wide-ranging applications as controlling construction robots, creating computer programs, and even designing molecular models.

[David Kahaner is on assignment with the US Office of Naval Research. He generally comments on activities in the Far East for inclusion in the Software Report column. Since we felt readers would be interested in a detailed description of virtual-reality trends in Japan, we also offer this special report. His comments are his own; they do not express any official policy.—Ed.]

Japanese researchers are making significant strides toward the use of virtual reality for equipment operation. Unlike devices that perform two-dimensional movements such as mice or joysticks, virtual reality employs ordinary human motions and spatial movements in the operation of equipment. Nippon Electric Co., Ltd. (NEC) is researching virtual reality's possible role in a three-dimensional computer-aided design system, while the Tokyo Electric Power Co., Ltd., (Tepco) is working on a system that uses this technology to analyze software. Tokyu Construction Co., Ltd., wants to adopt such a system for the remote-control operation of construction robots. Fujita is conducting research jointly with the US company VPL on an application that can operate construction robots via communications lines.

Virtual-reality technology uses computer graphics, 3D images, and similar tools to make things that do not actually exist appear to the operator as though they do. That way the operator's own

hand or finger motions can move an object around, or alter its shape, in virtual space. In progress is research that aims to create such objects inside virtual space rather than using 3D CAD shape models or actual construction robots.

A robot, of course, moves in three dimensions, and 3D CAD obviously uses 3D space. Virtual reality enables all kinds of operations to be done via human movement. The system links a "hand" or "arm" inside "space" to the 3D movements of the machine or model being manipulated. Such an arrangement facilitates intuitive comprehension of what is happening. Researchers hope to use this technology for handling 3D CAD and manipulating construction robots.

CAD system input provides a good example of this concept. If we want to enter a flat drawing, a mouse is perfectly adequate. Much more is involved, however, when we want to enter a 3D object. A mouse moves easily on a flat surface, but not so in 3D space. A mouse simply cannot give operators a true sense of working with a 3D representation. Researchers are now looking at the use of virtual reality to incorporate 3D movements and to develop input interfaces that allow intuitive operation.

Transforming models with "data gloves."

A virtual-reality system developed by NEC's C&C System Research Laboratory gives operators a feeling that their hands are actually touching the 3D CAD shape model they are manipulating. This

system easily performs operations that make it seem that the model is made of clay. Users can change the model's shape. They can add or delete parts using hands (agents) that appear on the screen. Virtual reality links the movement of these "agents" to those of data gloves the operator wears. The operator "holds" the model, or so it seems—a sensation no mouse can impart.

The NEC system uses data gloves manufactured by VPL for the input device. This glove-shaped device gathers hand-position and finger-bend data through magnetic sensors and optical fiber. A liquid-crystal shutter glass system made by the US company Solidray Lab helps create a 3D image. Liquid-crystal shutters produce the image by rapidly changing between left-eye and right-eye images.

On the screen itself appear the model, the hands that manipulate the model, and the operator's hand agents. Every operator wears data gloves on both hands, so two hand agents appear on the screen for each operator. The hand agents grip the model, move it about, and change its attitude. When linking two terminals so that joint manipulations can be performed, four agents appear on the screen. Triangular and rectangular functional icons for shifting the perspective, changing the color, or otherwise manipulating the image appear at the top of the screen.

The operator uses a planar cursor to change the model's general shape. When the shapes of the hands wearing the data gloves are altered, the agents appear as panels rather than hands. This panel is the planar cursor. Through hand movements, the operator can move the planar cursor to any position or attitude. The operator can also cut or sculpt the model in this plane, thereby shaping it.

The planar cursor represents positions in 3D space and also the inclination of the plane in that space. Manipulating the planar cursor thus is very difficult with a mouse that moves only in two dimensions. With data gloves, however, operators need only tilt their hands to put the model in the proper attitude for cutting. Such processes are intuitive operations that require no memorization of special procedures.

Touching a yellow plate with a hand agent calls up a toolbox. The toolbox contains prefabricated shape models and parts. The operator can pull out a shape model from the toolbox and compare it with the model being designed.

Suppose we want to add tires to the model body of a car. "Three-dimensional computer graphics alone is not adequate for getting the right positions," says Shoji Kawagoe, a research section chief in NEC's Terminal System Research Department. "You need something that gives you a 3D view." Here again, we are talking about incorporating an intuitive sense into the work, similar to what we have when building a plastic model.

By altering one's hand gestures, an operator can also change the agents to a pointer shape. This pointer then can designate a specific part of a model that is being studied by more

***In virtual reality, the operator
"holds" the model, or so it
seems—a sensation no mouse
can impart.***

than one person. Touching the green plate calls up a coloring function called a "color ring." After placing the model in the middle of the color ring, the operator gives it the desired color by designating that color with the pointer.

High positional precision. "We developed this system to use 3D CAD in a network environment," says Kawagoe. By connecting remote sites via communications lines in a system, a number of people can simultaneously study the same CAD model. The peculiar sense of immediate "presence" virtual reality affords makes it seem that the model is actually there with the operators as they manipulate it. "We plan to move our development work in the direction of further enhancing CAD functions."

For serious use, however, data gloves do not provide adequate precision. They can detect positions in 3D space by means of magnetic sensors, but this detection is only precise to within several centimeters. Nearby metal objects also readily influence precision. Adjusting the system, therefore, is very time-consuming. The data glove is a simple device in which optical fibers and sensors are installed in a glove. Its simplicity helps explain why it is now used so widely in virtual-reality research. For future practical applications, "the poor precision afforded by data gloves is a system bottleneck relative to CAD needs," according to Kawagoe. A new spatial positioning input device therefore is needed.

Recognizing gestures with image processing. Even though various manipulations can be performed merely by hand movements, using a special device like data gloves that must be worn is nevertheless troublesome. Much better would be accessing virtual-reality environments without such an annoyance. Researchers at ATR's Communication System Research Institute (Kyoto) are working on a system that uses image processing to recognize facial expressions and gestures. Such a system would provide a more natural interface that could be used without wearing any kind of special gear.

The ATR system employs images captured by two overhead charge-coupled device (CCD) cameras. The system abstracts an image of the hand, then charts the positions of the center of gravity and of the fingertips. These measures reveal the shape and position of the hands. The distance from the center of gravity to the fingertips tells whether or not the fingers are bent.

***Researchers are also studying
virtual-reality input in
applications involving computer
program design.***

The broadness of the space in which the positions must be detected determines the positional precision. If the hands are to be moved within a cubic area 1 meter on a side, a precision of about 5 cm—several percent of one side—can be achieved. "Currently, the positional precision is about the same as with the data gloves," says Haruo Takemura, a researcher working in ATR's Knowledge Processing Lab.

Recognition based on multiple signs is built up by combinations of finger-number and finger-bending patterns. Distinguishing between individual fingers, however, is difficult. The system can tell a thumb from a forefinger, but not a forefinger from a middle finger. Using a Sun 3/260 system as the host permits four recognitions every second. Recently, this recognition speed has been increased, and a speed of 15 recognitions per second is now available.

Currently, all this sign recognition is based on static signs, but Takemura speaks of a "desire to move development efforts toward recognizing gestures" from continuously changing "dynamic" sign patterns. If continuous changes can be recognized, ATR researchers believe that it will be possible to distinguish individual fingers.

ATR is also actively investigating line-of-sight detection and detecting facial characteristics. The goal is to combine these techniques so that an object can be manipulated in virtual space by image processing alone, without the operator having to don special gear.

3D Imaging of software structures

Researchers are also studying virtual-reality input in applications involving computer program design. Tepco's Systems Research Lab is investigating a system that represents intertask relationships during program execution as 3D images. The intertask relationships are studied as one manipulates the 3D images by hand while wearing data gloves. Features of this system fall into categories of visual analysis and visual design. Visual-analysis features help extract bugs and analyze task structures. Visual-design features help convert the results of modifying the program to source code. The intuitive manipulation made possible by virtual reality plays a major role in the system.

In visual analysis, the system first extracts trace data from a program that is running and renders it visible as a solid 3D

image. The trace data consist of information on the internal operating conditions when a program is running. Such data include a particular task's start and stop, and the files or tasks called up. The system displays the image on a large back-projection display and makes it three-dimensionally visible with liquid-shutter glasses. To analyze the program, extract bugs, and adjust the program, the operator uses data gloves to grip and move the tasks or files represented as rectangular parallelepipeds in 3D space. By appealing to the operator's intuition, this method makes it easy to analyze or modify programs.

Using actual trace data, the system renders the operational status of the control programs used in Tepco's central control facility three-dimensionally visible. A thick white line in the middle of the display represents the time axis. The operator calls up files and tasks in the order in which they are connected by the other lines, thereby indicating how the program is operating. By watching this screen, the operator can easily see the timing with which tasks are run and files are called up while the program is executing.

To modify a program, the operator uses the data gloves to move or add internal structures inside the program as it is being represented three dimensionally. Tepco has given the system 13 modes in which the data gloves can pose. The modes are used to enlarge or reduce the image on the screen and to move the figure in the screen. Commands are also available for opening menus and making selections. The operator uses these features to modify a program. The operator grips tasks with an agent, for example, and moves them back and forth along the time line to alter their operational timing. The operator can use these procedures to insert a completely new task. The system immediately performs a simulation to determine the effects of changing the task positions and updates the screen display.

In the past, somebody had to read numerical trace-data outputs and create a line diagram (like a train routing diagram) before a control program could be studied. To study the effect of altering the call-up timing, the lines had to be redrawn. A transformer station control program contains more than 20 tasks and 20 files each, making a line diagram on paper very complicated. Looking at such a diagram and understanding it immediately is difficult, and redrawing the lines can be tricky.

Tepco's system not only automates such analyses but also uses virtual reality to make the analysis three dimensional and easier to manipulate manually. These are very attractive features. By imparting a virtual shape to a program that is not a physical object, human intuition can be employed with a sense that one's hands are moving the object "itself." This approach should make such analysis work more efficient.

"We will finish the interface portion this year," says Mikio Okada, a lead researcher at Tepco. "We then hope to get to the practical level within two more years." With the visual-

analysis functions of the same system, complex relationships between tasks and files within a program can be represented in a form that humans can readily comprehend. This technology should have applications in process control systems in production lines as well as in software research.

Research now has also moved ahead to visual design. Visual design, according to Okada, is "a feature for expanding a picture in software." Visual design takes the 3D figures or diagrams analyzed by visual analysis and converts them to source code. The goal is to achieve a system that takes the results of manipulation to figures or diagrams and directly reflects these in programs.

Past research focused on generating program text code from symbols. This research, however, was limited to lower command levels, even within a program structure. Evidently, little research has been done on taking the large program structures constituted by task relationships and transforming these from symbols to source code.

Now, the call relationships between tasks, or between tasks and files, can be converted to code. Next comes research on encoding chronological relationships and call-up timing. "It's very hard to properly reflect wait times," says one researcher.

When used in visual analysis, the lack of precision of the data gloves has little effect. In visual design, though, this imprecision becomes a problem because time relationships must be expressed precisely at positions in 3D space. Researchers hope they can reflect time relationships in conjunction with the visual-design features this year.

Remote control of construction robots

Research on input systems employing virtual-reality technology is not limited to CAD or other program software. Research is also being conducted in hardware fields such as robotics. Tokyu Construction is studying a system for possible use in the remote control of deep-foundation work robots (a type of construction robot). Researchers are seeking to control construction robots with 3D images and hand movements. This technique is called tele-existence.

When an opening is too small to permit the entry of construction equipment, workers must dig holes to put the foundations in place. These are called deep foundations. Robots have been developed to replace humans in this deep-foundation work. After the hole is dug out, workers line the excavated surfaces with steel plates called liner plates, and pour in the foundation concrete. Accordingly, the diameter of the excavated hole must be larger than the diameter of the foundation put in place by the thickness of the liner plate.

With remote control, the positional relationship between the robot bucket and the excavated surface must be known accurately. Tokyu compared the work precision 1) when using images from one camera, 2) when using 3D visualization employing two cameras, and 3) when relying on the naked eye.

Three-dimensional visualization of images taken with two cameras provides roughly the same degree of work precision as the naked eye.

Results showed that the sense of positional relationships in space with 2D screen images taken with one camera is inadequate, and that 3D visualization is necessary. 3D visualization of images taken with two cameras provides roughly the same degree of work precision as the naked eye.

The remote-control system now being developed achieves 3D visualization by using two CCD cameras to take a stereo-visual image. To create the 3D realization, half mirrors help synthesize two CRT screens equipped with Polaroid filters. A 3D visualization system is configured much more simply this way than when using a head-mounted display or the liquid-crystal shutter technique that switches the image between the two eyes at high speed. "We can obtain good pictures at low cost," says Masayuki Takasu, director of Tokyu's Mechatronics Development Lab.

Electrohydraulic bilateral control gives "feeling" during manipulation. A dedicated position input device moves a bucket when excavating. For this position input device, the arm of a small experimental shovel is shortened to one-fifth normal size. To move the shovel bucket to the designated position and perform excavation work, operators manipulate the input device that they hold while watching a 3D image. For the position input device and the small shovel, the electrohydraulic bilateral control system has an electrical system on the input side and an hydraulic system on the output side.

With conventional lever control structures, the operator controls the valves of the hydraulic actuators. However, there is no immediately apparent relationship between the vertical and horizontal movements of the levers, on the one hand, and the movements of the arm, on the other. Operating the arm and bucket correctly with the levers then takes considerable skill. With this new method, however, a beginner supposedly can excavate in the desired location. Lever operation requires considerable skill to accurately determine the excavation position. The new method easily makes precision determinations of the position.

This system comes with load cells on both the input and output sides to give the operator "feeling" when excavating, thereby appealing to the tactile sense. Precious little research is being done on virtual reality's role in this sense of touch.

"We have a situation in which a construction robot in America can be controlled from Japan."

"It's a big help on the job just to be able to feel some kind of feedback when a force acts on the tip of the bucket," explains Takasu. With "feeling," operators can tell when they have struck a hard layer, so this sensory feedback is absolutely necessary for deep-foundation robot operating systems.

Properly adjusting the load cells makes use of force feedback difficult. Obviously, the full force acting on the bucket cannot be returned to the operator—such force would be far too strong. Some appropriate fraction of this force must be returned, after a proportional reduction is made. "It is very difficult to find the best proportionality because of the differences between individual operators," says Takasu. "The fatigue factor will be too great if the force returned is too large. But there will not be enough feeling if the force returned is too weak." Now that researchers have some degree of positional precision within their grasp, they are now working on ways to determine the optimal operating-force feedback proportion while increasing the number of samples.

Long communications links to construction sites. Fujita is also conducting joint research with VPL on applications of virtual-reality technology for construction robots. The goal again is to develop remote-control systems. More specifically, however, the two companies want to connect domestic or overseas construction sites via ISPN (integrated service digital network) lines to control construction robots from a control center and to monitor and control site conditions. Fujita is primarily responsible for developing the hardware for the construction robots and communications technology, while VPL is developing the software for the computer graphics and virtual-reality technology used with the robots.

The system now being tested synthesizes camera images and computer graphics, and presents them on a 3D display that the operators wear. The video images give operators a sense of being "right there" at the construction site. They can manipulate the robot easily with cursors or pointers displayed with computer graphics. The overall system is divided between Japan (Fujita) and the United States (VPL) so that researchers can experiment with remote-control operations over extremely great distances. "We have created a situation in which a construction robot in America can be controlled from Japan," says Kenichi Kawamura, in charge of the project for Fujita Research.

"The biggest task problem facing us now is the time delay involved in long-distance communications," says Kawamura.

This is not much of a problem when both the construction site and the control center are in Japan. When one is in Japan and the other in the United States, however, there is a delay of a second or more from the time the command is given until the robot actually moves. The same is true when sending a camera image from the robot. Fujita hopes to cope with the delay problem by employing computer simulations of the robot movements and displaying predicted motions on the operator screen.

"Being there" with camera images

Besides virtual reality, there are other possibilities for creating easy-to-understand interfaces. Real camera images can also be used to impart a sense of "being there" and thereby enhance operability. Hitachi has developed a prototype of a plant monitoring system that uses camera images to provide such a sense of "being there." This system does not use 3D visualization or other virtual-reality technology, but does give a real sense of plant work-site conditions and thereby seeks to enhance operability.

The prototype system uses a model of a monitoring system for a thermal power plant. Plant conditions can be monitored and controlled by directly manipulating the camera images. Suppose, for example, that the operator wants to check on the condition of the fire under a boiler. The operator uses a mouse to move a pointer and clicks it over the image of a boiler peephole. A frame appears on the screen, indicating that a peephole has been selected as the device. By clicking the peephole again, the operator switches the screen to an expanded image of the peephole. Clicking the mouse a third time changes the screen to the image of the flames that a monitoring camera is taking. Simply indicating it on the screen enables the operator to see a desired view, making operation extremely easy.

When changing the operational status, furthermore, the operator calls up the operating panel image by using the mouse to designate the image of the device. Once the image of the control panel appears, the operator can designate its dials with the mouse. Moving the mouse moves these dials. The actual dials on the control panel at the worksite will also move as the mouse is moved, thereby changing the operational status of the equipment. The operator can even listen to the sounds of the worksite and make decisions on the status of the system based on these sounds.

Force feedback experiments

"Until recently, most interaction research has focused on the visual and auditory senses," explains Nadao Iwata of the Structural Engineering Department at Tsukuba University. "Research on the tactile senses of touch and pressure has lagged behind. To enhance human manipulative sensation, however, we need to study the physical interactions involving immediate bodily sensations as well as indirect sensations."

If we run into a wall, we bounce off of it. How far can we go in reproducing these sensations of touch and force, so common in our everyday experience, in the realm of virtual reality? According to Iwata, there are four ways in which dynamic feedback can currently be expressed in virtual-reality technology.

Master-manipulator technique. Here, a manipulator based on mechatronics technology communicates a physical reaction back to operators as they perform actions in virtual space. This equipment often involves large-scale hardware and is very expensive. Usually it is considered unsuitable for human interfaces.

Wire (tension) utilization. This method detects positions from the length of wires. Controlling those wire lengths produces physical reactions. Small movements such as those made with the fingertips can be used to good effect. The range of movement is limited, however, because the wires stretching in all directions may get tangled.

Joystick. This input/output system is made up of a control stick that can be freely tilted longitudinally and laterally, and equipment that detects the angle, direction, and force of the tilts. Its advantage of desktop compactness is somewhat offset by the limited number of degrees of freedom.

Data gloves. Research is progressing on feedback from gloves that use shape-memorizing alloys and air-pressure cylinders. Data gloves are limited to what can be felt with the hands, however, and cannot express the sensation of running into and bouncing off a wall.

One of Iwata's experiments involves a feedback system based on the master manipulator approach. Ordinarily, a mammoth system comes to mind at the mention of a robot arm, but the apparatus under development at Iwata's laboratory is a very compact desktop system.

It consists of a manipulator unit that follows hand or arm movements with six degrees of freedom and three actuators that follow finger movements. Operators can move their hands and fingers independently. When sensors detect the position and movement of the hand, a virtual hand in a monitor screen moves accordingly.

When the hand in the screen strikes an object, the mechanically controlled manipulator motor restricts the movement of the operator's hand, producing a real sense of resistance. Similarly, when a virtual rubber ball is grasped, the operator's hand senses elasticity, and when an object is lifted, the operator senses the "weight" of the object.

For visual sense, rather than using a head-mounted display, this laboratory plans to project images just as they are on a high-resolution screen. Such an approach would result in the operator's hands fixed to the manipulator coming into the field of vision. Consequently, a mirror is placed in front of the operator's face at an angle of 45 degrees and the computer screen is projected there. The mirror then hides the hands from view.

***Recreating the trial-and-error
environment of a design process
in a virtual-reality environment
will sharply reduce the time and
effort spent making a mock-up.***

"If we can recreate the trial-and-error environment of a design process in a virtual-reality environment, the time and effort spent making a mock-up could be sharply reduced," explains Iwata. "Take the case of automobile design. The designer has to make small models out of clay. In giving form to an inspiration, not only must the work be seen, but the sensation of the hands is also very important. If this kind of intuitive expression could be input, the computer could handle the tedious work of the design process."

Iwata's laboratory is conducting experiments in which virtual hands handle a single-lens reflex camera created on a computer-graphics screen. The hands can determine the camera's weight balance and show how it feels when operated.

Upstairs, downstairs. Another large-scale virtual-reality project in development at Iwata's Tsukuba laboratory is the virtual perambulator. Iwata wanted to build an apparatus that would take the previously separated bodily sensations and handle them together. With this apparatus, the walker's upper body is fixed in place and a head-mounted display is placed on their head. Image sensors detect the position of the head. Attached to the walker's toes are ultrasonic wave generators. Measuring the time required for the ultrasonic waves to reach receiver units placed in three different locations determines the positions of the walker's feet. These motion data are sent to a computer that generates a virtual space, and the view through the head-mounted display changes in real time in response to the walker's movements.

The perambulator is made so that the tension on wires attached to the feet produces the sense of reaction or resistance associated with climbing or descending stairs. When the walker ascends a step, the wire length is regulated so that the take-off foot feels the force of resistance. To represent the reaction force involved in opening a virtual door, the virtual perambulator includes a manipulator having six degrees of freedom. Once this system is perfected, designers of large structures, city developers, and public park constructors can perform simulations to find out ahead of time what it feels like to walk around in such facilities.

"The demand for view assessment is bound to continue to grow," says Iwata. "Computer graphics are already being used,

"Students of shodo (Japanese calligraphy) could use this [force feedback apparatus] to trace the pen strokes of the master. Wouldn't that be great?"

but, with this apparatus, bodily movement is coordinated with visual images. A more lifelike model can be experienced. I think it might have useful applications in simulating living environments. A new house could be experienced while it is still in the design stage, for example."

A craftsman's fingertips. Yukio Fukui, a researcher with the Product Science Research Institute operated by the Agency of Industrial Science and Technology (AIST) under the Ministry of International Trade and Industry (MITI), has developed a force feedback apparatus that uses an XY recorder. For some time, researchers have used these recorders—useful for recording changes in coordinates relative to vertical and horizontal axes—to record vibrations or temperature fluctuations. Fukui decided to apply them to the field of virtual reality. His research falls into the "joystick" category.

The head unit in the XY recorder is equipped with four-directional strain gauges. Moving one of the heads by hooking a fingertip into a depression in the middle of a gauge consequently moves a cursor on a screen. As one tries to follow the object in the screen, the feeling is no different than with a conventional mouse. But when the cursor tries to bite into the object, the XY recorder stops dead in its tracks.

"The force and direction of the fingers are measured with the strain gauge. Information on the position of the cursor on the screen is fed back into the XY recorder, and the finger movement is restricted as though someone said, 'That's as far as you go,'" explains Fukui. This apparatus has three advantages: It can move freely, without any constraints of particularity, so long as movement is limited to the two *X,Y* dimensions. Force control and computation is simpler than with robot arms and wire tension because the positions are determined by two *X,Y* coordinates. Finally, the recorders used are commercially available.

"Sometimes you have an object with a curved surface that is only slightly deformed in one place," explains Fukui when asked about following subtle shapes in a virtual environment. "The deformation might be undetectable to the eye, but readily discernible to the touch. When making a metal mold or die, skilled craftsmen might feel it with their fingers and detect subtle curves that they then correct. This kind of

trial-and-error operation can be reproduced in a virtual-reality environment."

In other words, the computer lets the operator feel subtle tactile sensations that cannot be expressed in words or discerned visually.

"We can think of many applications in the field of education and training," says Fukui. "Students of *shodo* (Japanese calligraphy) could use this to trace the subtle pen strokes of the master. Wouldn't that be great?" Maybe this portends a future in which educational or training software will be sold not just for the visual (video) and verbal (audio) information it imparts, but also for the "feeling" of the master that it enables the user to experience.

Fukui plans to start development work on other 3D feedback devices. "We are now developing a system having a stick that moves longitudinally, laterally, and vertically. Moving something like a virtual spatula, the stick will trace out or transform shapes."

As Iwata sees it, the history of the human race could be likened to a series of attempts to broaden the "virtual world" (imaginative world) and thereby expand human awareness.

"Take the ordinary telephone," says Iwata. "You truly experience this expansion of the virtual when you listen to the voice of someone speaking on the other side of the planet. The same can be said of sitting in your parlor and watching things happening on the surface of the moon. In like manner, virtual reality has the potential for explosively expanding our world of awareness."

It is amazing to see the world of data inside a computer spread tangibly before your eyes by virtual reality, and thrilling indeed to experience the initial moment of entering the imaginary world that is created. Virtual-reality technology continues to make all kinds of "impacts" on us.

Now, the sensory information fed back to us in real time from the world of virtual reality presents new world vistas to us. We can visit the interiors of virtual buildings, delve into the world of molecules, and experience the feeling of taking action and doing work. This process gives us an intuitive understanding that heretofore had been inaccessible.

"What to a human being is the sense of reality?" we ask ourselves. As we work out the answer, we may be incorporating not only the tactile senses into the world of virtual reality, but the senses of smell and taste as well. Technology will continue to expand the world of our experience and awareness, without limit, until it seems that another world, a new earth, has fallen into our hands.

Spidar

Makoto Sato, a professor at the Precision Engineering Research Laboratory of the Tokyo Institute of Technology, is the developer of Spidar (Space Interface Device for Artificial Reality), an acronym that also suggests a "spider."

To use this device, one first sticks one's index fingers into

thimblelike rings. Each ring is suspended from all directions by wires. Yanking down on the ring changes the length of the wires. A rotary encoder measures these changes by detecting the position of the finger.

Anaglyph-based stereoscopic vision provides the visual input. By looking into red and green glasses, the operator sees a 3D wireframe jar in a large screen.

When I tried pushing the virtual jar with my finger while it was still in the thimblelike ring, the cylindrical jar caved in. It took on the form of a large, squarish peanut or gourd. My finger, meanwhile, was locked into place and could not be moved. I really felt as though I had bumped into something.

"When the finger reaches the position occupied by the object, the wire is restrained and your motion is restricted," explained Sato. "This reproduces the sensation of touching something."

Next I tried pushing on the bulges in the jar with my finger. The bulges gradually caved in, and the jar returned smoothly to its cylindrical shape. The jar has a constant volume, so pushing on one place will bring the depressions back out. It feels like making a clay jar on a potter's wheel.

This unique apparatus developed from a very simple wish. Sato explains. "Seven years ago, I saw the Fujitsu pavilion's 3D video at the science expo. I was extremely impressed. In one scene a DNA (deoxyribonucleic acid) double helix twists its way up toward the ceiling. It was so real—I wanted to reach out and touch it."

Sato thought how great it would be if only one could "touch" a virtual object—if he could only use his hands to move and change the shape of a physical object. Sato's idea became a reality just three years later. At another science expo in 1988, he unveiled what might be called the predecessor of Spidar.

This prototype apparatus suspends a ball from all directions by wires. When a billiard stick strikes this ball, changes in the lengths of the wires detect the motion of the ball. These data are input into a computer while the virtual ball rolls and strikes another ball.

Through this "virtual billiard table" Sato realized his dream of "touching" an imaginary object in virtual reality. The next problem was to control the reaction feedback from the object touched. It took Sato four more years to create Spidar. Spidar II, currently under development, uses both the index finger and thumb.

Sato calls his Spidar II the "world of virtual building blocks." In this system fingers are inserted into two rings. Each ring is suspended from four wires, making eight wires in all. When operators look through 3D glasses equipped with liquid-crystal shutters, they see a stack of building blocks with holes in them and a stick passing through their centers.

"Try picking up a block with the thumb and index finger and moving it to another stick," said Sato. At first, judging the distance is difficult, and one waves about in thin air. The blocks are hard to grasp. When a block is grasped, the op-

***We can visit the interiors of
virtual buildings, delve into the
world of molecules, and
experience the feeling of taking
action and doing work.***

erator can immediately "feel" the object from the restraint on the wires. I had a difficult time transferring the block to another stick, and was relieved when I finally got the hole in the block on the stick so I could let the block drop.

Virtual reality in molecular design

The range of applications for this technology is very broad, extending past the CAD field to education, medicine, and entertainment. "It is also useful in molecular design in such fields as pharmaceuticals and protein synthesis," says Iwata.

"Suppose, for instance, that you wish to make a new molecule by means of molecular bonding. You have to study the attraction and the potential that exists between molecules and find out where the low-potential, readily bondable locations are. In the world of giant molecules that have complex structures, however, the researcher is faced with a myriad of parameters and conditions that exceed the processing limitations of computers. So, what do you do? You build a molecular model with computer graphics and then use virtual hands to manipulate and join the molecules." □

Questions regarding this column can be addressed via e-mail to David K. Kahaner, US Office of Naval Research, Far East, at kahaner@cs.titech.ac.jp.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate numbers on the Reader Service Card

Low 171

Medium 172

High 173



Richard H. Stern

Oblon, Spivak,
McClelland, Maier &
Neustadt, P.C.

1755 Jefferson Davis
Highway, Suite 400

Arlington, VA 22202

A back door to protecting look and feel?

After the Second Circuit federal appeals court's decision in *Computer Associates International, Inc. v. Altai, Inc.*, last year, it appeared that the door was shutting on copyright law protection for the "sequence, structure, and organization" (SSO) of computer programs, alias "protection of nonliteral program elements" alias "look and feel."

For convenience, I will refer to all of these, collectively, as "look and feel." But they may be distinguished from one another. SSO is the selection of variables, partitioning of routines, and other determinations of how to code a program. Look and feel is the visual appearance of a user interface and the keystroke patterns used to initiate tasks and functions. Nonliteral elements of computer programs generically comprise any aspect of a computer program, including the foregoing items, other than the actual lines of code in the computer program.

The Second Circuit scoffed at the 1986 decision in *Whelan Associates, Inc. v. Jaslow Dental Laboratory, Inc.*, which had ushered in the era of look and feel. The court said that the *Whelan* decision had an "outdated appreciation of computer science." *Whelan's* approach, the *Altai* court said, should be replaced by a "filtration" analysis. Under the filtration approach, a court begins by filtering out of the copyrighted computer program all material that is in the public domain, that consists of stock elements, or that is dictated by "external factors," efficiency, or functional considerations.

(The court mentioned five so-called external factors: the mechanical specifications of the target computer platform, compatibility requirements of other programs, computer manufacturer design standards, user industry demands, and generally accepted programming practices.)

The court then compares the resulting filtrate with the accused computer program of the defendant for "substantial similarity." Only if the filtrate and the defendant's work are substantially similar can the court find copyright infringement.

Other courts then began to adopt the *Altai* test and to forswear *Whelan*. Even the judge from the *Lotus 1-2-3* case tried to climb on the bandwagon, asserting that his approach was "consistent with" that of *Altai*. What does all of this mean, in terms of who prevails? After you filter out all of the copyright-refrangible, utilitarian detritus (usually recognized by its acronym), what has passed through the filter, and what is left behind? It seems extremely unlikely that any SSO, look and feel, or the like will pass through the filter. If *Altai* becomes the general rule of copyright law, therefore, look and feel will receive scant protection under copyright law. Will that mean the demise of look and feel?

Not necessarily. In December 1992, the Second Circuit granted a petition for reconsideration of, and then rewrote, one part of its ruling in the *Altai* case. As just indicated, the main part of the opinion had dealt with copyright issues. One portion of the opinion, however, addressed claim by the plaintiff copyright owner CA.

CA claimed that the defendant Altai had misappropriated trade secrets in the computer program in violation of state law. A former CA programmer (Arney, not a defendant in the case) had left CA and gone to work for Altai. Arney, without Altai's knowledge, had taken a considerable amount of CA computer code with him when he left CA, and he incorporated it into one of Altai's computer programs. When Altai learned of this, it caused the computer program to be rewritten into a form that the court of appeals held did not infringe CA's copyright. The

court reasoned that the only elements of substance that the rewritten program took from the original program were unprotected under copyright law (because they were dictated by efficiency and similar considerations).

The court of appeals initially held, 3-0, that the trade secret claim was "preempted" by the federal copyright law. The court said the same challenged acts of Altai, as CA proved them at trial, constituted both copyright infringement and trade secret violation. This view required that the state law be considered superseded by the federal copyright law. The reason was that section 301 of the Copyright Act does not permit state laws to apply where they duplicate federal copyright law.

When the court of appeals reconsidered the trade secret preemption issue, it held, 2-1, that the copyright owner's claim was not preempted, in the circumstances of this case, and therefore it remanded the trade secret aspect of the case for a trial of the merits. It ruled that preemption does not occur when two factors are present. The first is that the stated cause of action requires an "extra element" of conduct, not required for the federal copyright cause of action. The second is that the extra element qualitatively changes the "nature" (not just the "scope") of the action. The extra element must be more than awareness or intent, the court said, because they merely alter the scope of the action, not its nature. If those are the only alleged extra elements, the state cause of action is preempted.

Here, the trade secret claim could have been based on illegal acquisition of the secret material or on illegal use. The trial court had held that the copyright owner's right to be free of trade secret misappropriation through use of the secret material was indistinguishable from its right of legal protection against copyright infringement by unauthorized reproduction and distribution of the copyrighted work. The court of appeals said that this was partly right:

The district court correctly stated that a state law claim based *solely* upon Altai's "use" by copying, of [the program's] nonliteral elements could not satisfy the governing "extra element" test, and would be preempted by section 301. However, where the use of copyrighted expression is simultaneously the violation of a duty of confidentiality established by state law, that extra element renders the state right qualitatively different from the federal right, thereby foreclosing preemption under section 301.

Here, the court of appeals said, are both wrongful use and wrongful acquisition issues, but the trial court did not fully address the issue of wrongful acquisition. The court found that Arney clearly breached duties of fidelity that he owed to CA, and that Altai may also have done so in receiving information from Arney and utilizing it. Altai rewrote its computer program once it learned of Arney's misconduct, thus terminating the copyright infringement. But nonetheless Altai may have acted wrongfully under trade secret law in rewriting and marketing the revised computer program, after learning that Arney took CA's trade secrets.

First, when it initially acquired the computer program, Altai perhaps "should have" known (because a hypothetical reasonable person would have known, in the circumstances) that the program had been stolen. That fact could have tainted any subsequent use of the information by Altai.

Second, Altai may have included CA's trade secrets (look and feel) in the rewritten version of the computer program. "Since Altai's rewrite was conducted with full knowledge of Arney's prior misappropriation, in breach of his duty of confidentiality, it follows that [the second Altai program] was created with actual knowledge of trade secret

***Look and feel will
not be protected
by copyright law,
but it may
nonetheless be
protected by
trade secret law.***

violations." That could have been wrongful under the theory that, after notice of trade secret rights, even an innocent "second user should cease the use" or else become "liable for damages arising from such use subsequent to notice." Because the trial court did not consider these issues, the court of appeals held, the case would have to be remanded for trial on this matter.

A trade secret claim could not be based on Altai's mere use of elements of the computer program that the copyright law left unprotected, because that would violate the principle of preemption. But such a claim could properly be based on Altai's improper acquisition of such unprotected elements, followed by use. "[T]hat extra element [use with notice of misappropriation of a trade secret] renders the state right qualitatively distinct from the federal right, thereby foreclosing preemption under section 301."

According to the revised opinion of the *Altai* court, therefore, look and feel will not be protected by copyright law, but it may nonetheless be protected by trade secret law. Does that mean that we are condemned to rehearse the whole miserable look and feel charade all over again, under a new caption—state trade secret law? Not necessarily; that depends on several things.

One is the extent to which courts will find trade secret law applicable to

look and feel aspects of computer programs. If a program is openly marketed, its look and feel may well be considered no longer secret. The look and feel may be discernible from the face of the computer program and therefore no longer secret if let loose by unrestricted distribution. (Such things might remain secret, however, after only a restricted distribution.) Further, the circumstances attending a second comer's acquisition of a computer program, or information about its workings, may or may not involve a duty of confidentiality toward the creator of the program. (See the box.)

Finally, there may be a major flaw in the whole idea of trade secret protection for look and feel, which the *Altai* court did not address. The features of CA's computer program that the *Altai* court held unprotected by copyright law were unprotected be-

cause section 102(b) of the copyright statute provides that copyright protection for an original work of authorship shall not "extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery." Copyright law does not merely omit protection of this subject matter; it explicitly excludes it, in codification of the doctrine of Supreme Court's 1879 decision in *Baker v. Selden*. That decision holds that such things should not be protected by copyright law, but should be protected instead, if at all, under the patent law.

The question therefore arises whether section 102(b) merely provides that these things shall not be protected under copyright law or, rather, provides that they shall not be protected at all except under the patent law.

(The Supreme Court's 1989 decision in *Bonito Boats, Inc. v. Thunder Craft Boats, Inc.*, holds that exclusion of some subject matter from the patent law constitutes a determination that such subject matter shall be left open to public use and shall not be protected by state law. In effect, Congress made a decision that the public is better off if free competition applies to such things unless they are protected by patent law. There appears to be no comparable ruling under copyright law.)

If the *Bonito Boats* principle is intended by Congress to apply to copyright, state trade secret law should not be permitted to intrude into this federal scheme of allocating subject matter among patents, copyrights, and the public domain. If that is not intended, however, trade secret law has a legitimate role in protecting state-recognized interests in such subject matter, supplementing copyright law without interfering with its purposes and objectives.

How that question is resolved may determine whether look and feel will continue to be a basis for asserting proprietary rights against would-be competitors to limit their access to the computer program marketplace.

Is the source code for mass-marketed object code a trade secret?

Some software proprietors assert that when a computer program is distributed only in object code, the source code is a trade secret. They assert that disassembly is both copyright infringement and theft of trade secrets.

Others take the view, however, that disassembly of a computer program bought on the open market is neither copyright infringement nor a violation of trade secret rights. The Ninth Circuit in the *Sega-Accolade* case, and the Federal Circuit in the *Atari-Nintendo* case, indicated that disassembly, when necessary to learn how an interface works or necessary for some other legitimate purpose, is not copyright infringement.

It is questionable that sealing part of a product up in a black box (for example, epoxying an integrated circuit) before selling it will preserve trade secret rights. The same principle would appear to apply to object code and source code. When object code is licensed pursuant to an express agreement against disassembly, however, the situation becomes much more murky, and the assertion of trade secret rights may become more plausible.

Expedited delivery

is available to all members residing outside the USA, Canada, and Mexico. We invite you to take advantage of this service providing delivery of your magazine weeks earlier.



For information on this service and its cost, contact:

Expedited Delivery
IEEE Computer Society
10662 Los Vaqueros Circle
PO Box 3014
Los Alamitos, CA 90720-1264
USA
Phone: (714) 821-8380
FAX: (714) 821-4010

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 177 Medium 178 High 179



Standards and innovation

Do standards stifle innovation? Definitely—and I wouldn't have it any other way.

Perhaps that seems harsh, especially coming from the Micro Standards editor. After all, innovation is the introduction of something new, and in our culture, newness is practically a synonym for goodness. But innovation, per se, is really neither good nor bad. It's not enough to simply ask if something is innovative; we need to explore the value of the innovation.

Mark Twain put it another way when he said, "A man with a new idea is a crank until the idea succeeds." (Perhaps he should have said, "... *unless* the idea succeeds.") Innovation gains credibility from succeeding. However, success is a vague metric—and more quantification of the term is necessary. I explore some of these issues here and discuss how to add value to the term "innovation."

Value of innovation

I've devised an innovative (read "new") scale against which innovations can be measured. I call it the wrinkle scale. A "wrinkle," according to the *American Heritage Dictionary*, is an ingenious new trick or method; a clever innovation. I propose that you think of wrinkles as necessary innovations. On the other hand, a gimmick is "a trivial or unnecessary innovation, as a gadget." Both represent innovation, the only variable is the value of the innovation. To use a contemporary example, consider how recent innovations in cola drinks could be placed on this continuum. The invention of diet cola, at least in my opinion, qualifies as a wrinkle. On the other hand, the new "clear" colas are obviously gimmicks (Figure 1). I might note in passing that gimmickry has the added connotation

of deception, either intentional or unintentional.

The key to this discussion is that the term "value" must be defined. It is here that the real problems begin, since one person's wrinkle is another person's gimmick. In the sphere of computing, I suggest that wrinkles are those things that make a system more open, more available, more effective, more efficient, and more accessible to the users. On the other hand, gimmicks are often things that make a system better for a vendor—and damn the public.

Let's examine the implications of standards stifling either wrinkle innovation or gimmick innovation. As I've said, standards certainly stifle some kinds of innovation, but the kind of innovation that we should be concerned about protecting is necessary innovation: innovation on the "wrinkle" end of the spectrum. To the extent that standards stifle gimmicks, it's all to the good. So let me restate the question: Do standards stifle necessary innovation?

Wrinkle<—————>	Gimmick
(Ingenious, clever,	Trivial,
necessary)	unnecessary)
Diet cola	"Clear" cola

Figure 1. Wrinkle scale.

Interface standards

High-quality standards, as I discussed in my December column, are standards with proper form and function, those that meet the requirements of fitness for purpose. Such standards are interface standards, not implementation standards. An interface standard defines a common abstraction layer between neighboring regions. Above and below the interface layer is some-

Stephen L. Diamond

SunSoft, Inc..

Phone (415) 336-4190

Fax (415) 336-4477

steve.diamond@eng.sun.com

thing different—a phase change. The interface itself does not really exist; it makes possible the regions on either side of the interface. In contrast to the transcendent nature of an interface standard, an implementation standard defines a particular, concrete product. An interface standard allows, even encourages, innovation. An implementation standard stifles it.

Let's explore the differences between interface and implementation standards by way of an example. Posix (and here I'm speaking of the operating system work in Posix, not user utilities) provides for applications to access OS services, such as reading and writing files, through an application programming interface layer (API). Think of Posix as the mortar between two bricks: the top brick, the applications program itself, and the bottom brick, the OS code (Figure 2).

Applications program
Posix (open)
Operating system (open)

Figure 2. Interface standard.

Applications programs are free to do anything above the Posix layer, and operating systems are free to implement below the layer in any manner. Only a boundary layer is defined in the standard. Because it is an interface standard, a vendor can choose to implement a Posix-compatible OS in many ways by simply replacing the lower brick. And because the interface layer hides dependencies at lower levels, the designer may choose to make implementation-specific trade-offs between software and hardware as a means to optimize the system.

Let's contrast this with MS-DOS (Figure 3). MS-DOS also provides an API layer for applications programs; applications program vendors are free to implement their applications in any way, so long as they abide by the rules of the particular vendor's migration of the API. But the somewhat skewed parallel with

Posix stops there, because MS-DOS is not an interface standard. Rather, it is an implementation standard, that is, a particular product defined and owned by a particular vendor. OS developers are not free to implement below a stable API to meet their particular needs. Without an open API, and without access to the definition of the internals of the OS, nobody else could develop a plug-in, innovative file system, or a deterministic real-time scheduler, or lightweight processes, or threads, or a microkernel. In fact, nobody, other than the owner of the implementation standard could do anything below the API that is guaranteed to work in any strong sense. Such implementation standards stifle innovation.

Applications program
MS-DOS API (proprietary)
MS-DOS OS (proprietary)

Figure 3. Implementation standard.

A more physical example of an interface standard is the "foot interface" in automobiles. The foot interface defines the relative positions and functions of two or three pedals: the accelerator, the brake, and sometimes the clutch. The foot interface does not define the color of the pedals, nor the material from which they are made; those are elements of the implementation of the pedals. Because it is an interface standard, designers are free to innovate in the implementation.

In my old Lotus Elan, the accelerator pedal was arranged such that I could tap both it and the brake pedal simultaneously, allowing me to accelerate

quickly out of corners. If the foot interface had been an implementation standard, I might have been forced to live with the same pedal arrangements that Aunt Sadie had in her Stanley Steamer or Model T or possibly any of a large set of unique interfaces. The innovation here was the interface of the three pedals arranged in a set fashion. The underlying technology for brakes—whether disk brakes, drum brakes, or dropping an anchor—was left to the implementation. People buy the implemented features of an interface.

Avoiding the quick fix

I've suggested that innovations must be judged on their value, and that the innovations with which we should be concerned are the necessary wrinkles and not the gimmicks. Figure 4 summarizes the impact of interface and implementation standards on the wrinkle scale.

I've proposed that implementation standards do indeed stifle wrinkle innovation. Also quality standards—that is, interface standards—actually facilitate such innovation. Interface standards, by their abstract nature, also tend to discourage gimmick innovation, while implementation standards encourage it. Unfortunately, interface standards require significant planning, while implementation standards are frequently used as a quick fix to a poorly understood market problem.

Is my wrinkle scale a wrinkle or a gimmick? If it, and the concept of interface vs. implementation standards is valid, standards developers and standards users should look at standardization tools from an entirely different and more useful perspective—how standards impact customers and the market.

	Type of innovation	
	Wrinkle	Gimmick
Interface	Encouraged	Discouraged
Implementation	Discouraged	Encouraged

Figure 4. Interface/implementation effect on innovation.

Software Report



David K. Kahaner

US Office of Naval

Research, Far East

kahaner@cs.titech.ac.jp

Micromachines; high-tech odds and ends

Using micromachines for medical treatment has drawn growing interest in Japan lately. These tiny devices can perform such medical marvels as delivering exact doses of medicine to specific parts of the body and performing precise microsurgeries. Often modelled on organisms found in nature, micromachines will draw upon the fine processing technology of semiconductors for their manufacture, as well as protein engineering and atomic manipulation techniques.

On other fronts, Hitachi's quantum effect device will greatly speed calculations, leading to possible size reductions for future computers. Also, Mitsubishi has improved the target recognition capabilities of its autonomous robot, giving it a potential role in the retrieval of satellites. Finally, the Japanese have mapped out plans to improve coordination among industry, research centers, and universities for more effective high-tech research and development.

Medical micromachining

Interest is growing in the use of intelligent micromachines for medical treatment and other applications. Research leading to the production of micromachines is flourishing in Japan. There they see a close connection between micromachines and so-called intelligent materials. These materials are the building blocks for producing micromachines and their even smaller cousins we might call nanomachines.

In the world of micromachines, dimensions for basic parts range from 10 nm to 1 mm. Obvious applications for micromachines include medical treatment, postassembly precision machinery, and cases where precise work must be undertaken in special environments. The high pressures and vacuums involved in undersea and

space applications are especially appropriate in this regard. However, the most concrete use for micromachines, at least as the Japanese see it, is in medical treatment.

Reports show concrete images of micromachines being used in medical treatment. Drug transport vehicles are a prime example. Ordinarily, the entire body gets a drug dose, whether the drug is administered orally or by injection. This conventional approach limits the use of strong, potentially toxic drugs. Researchers would like to find a drug delivery system that can deliver medicine to specific sites in the body where it can act for specific lengths of time.

Microrobots provide another important application for this technology. A micro-mobile robot ought to be able to go through tubular cavities such as blood vessels or the spinal cord cavity. Surgeons then could insert a tiny laser endoscope into a diseased or damaged area. A small manipulator attached to the end of the endoscope could excise only as much of the affected tissues as necessary. This development could have a major impact on surgical techniques. Advances in micromachine technology would also lead to dramatic progress in the miniaturization, flexibility, and functions of artificial organs.

Size-related problems encountered in micromachine design include marked increases in friction, viscous resistance, and surface resistance, and a noticeable decrease in strength. Rigorous computer simulation then will be essential in the actual design phase of micromachines.

Micromachining—the technology used to produce micromachines—requires that we apply the fine processing technology used in making semiconductors. We will also need to develop methods that use protein engineering and atomic manipulation to assemble molecules and atoms.

One recent example of micro-machine production technology is the trial manufacture of cantilever-beam-type actuators. These devices mimic the ciliary movement of living organisms. Each actuator consists of a cantilever made from two kinds of polyimides having different rates of thermal expansion. Between the two sheets of polyimide is sandwiched a metallic resistance wire.

Residual stress initially forces the cantilevers to curve upward about 250 μm from the substrate. When current flows through the resistance wire, joule heating warms the cantilevers. Like a bi-metal, they bend down to the substrate. Distributing many actuators over the surface of a material endows the material surface with transport functions, thus creating a tiny carrier machine.

To find a model for micromachines in living organisms, the Japanese are investigating bacterial flagella. Known as nature's smallest "molecular motors," flagella use protein molecules as their components. To a bacteria, its flagellum acts as a screw or propeller. Recent reports cite experiments that freely controlled the rotational speed of flagella. Researchers fixed salmonella bacteria in a nutrient medium on the tip of a glass tube 1 μm in diameter. They applied a voltage between two electrodes set up on the inside and outside of the tube. The rotational speed of the flagella of the bacteria then was proportional to that voltage.

Considering the potential that micro-machine production technology offers, Japan's New Energy Development Organization began a Micromachine Technology project in 1991. As part of a large government project (the Large Industrial Technology R&D System), this 10-year R&D project will receive about ¥25B from the Ministry of International Trade and Industry (MITI).

Hitachi's QED

Hitachi recently announced a prototype quantum effect device that combines superconducting and semi-

conducting electrodes. The device performs simple calculations at 2.2K. It consists of two superconducting niobium electrodes separated by a polycrystalline silicon gate. One electrode is needle-shaped and is 0.08- μm wide. Separated by 0.01 μm , the opposite electrode is planar. Both are placed on a monocrystalline silicon substrate.

The device uses the Andreev reflection principle, a quantum effect occurring at the boundary between superconductors and semiconductors. The gate voltage (200 mV) controls the phase difference between waves passing between the electrodes. Hitachi hopes that research in this area could significantly reduce the size of future computers. They anticipate computation speeds of 10 to 100 ps per gate (equivalent to that of Josephson devices) with power consumption of microwatts per gate. *[For background info on QEDs, see IEEE Micro, Feb. 1989, pp. 96, 97.—Ed.]*

Melco's autonomous robot

Mitsubishi Electric Corporation's Central Research Laboratory (Melco) introduced an autonomous robot that selects and grasps certain objects from among a number of moving targets. The robot uses three neural nets. The first recognizes the shapes of articles it wishes to grasp, and the second analyzes the direction and speed of moving objects. The third controls a manipulator—the so-called "magic hand." This robot may be useful in recovering satellites and in deep-sea operations.

Melco previously tested an autonomous robot system that used two neural nets to analyze the position, speed, and direction of movement of a moving object. The nets also controlled the joints in a manipulator and extended an arm to grasp moving objects. By adding a third net, the lab has succeeded in having the robot recognize the shapes of objects it wishes to grasp.

Recently, the US Space Shuttle Endeavor crew walked in space to manually recover a satellite that had failed

to reach its proper orbit. On future missions, this newly developed robot might be used instead. After memorizing the appendage used for grasping the satellite, an autonomous robot could be sent out to retrieve it. The human crew then would not have to engage in such a dangerous activity. Satellites made at great cost can often be reused if only their fuel gas can be replenished. Hence there is strong demand for such recovery operations.

AIST bolsters research

Based on the substantial growth of regional high-technology industries in Japan, MITI's Agency for Industrial Science and Technology (AIST) will develop new regional technology policies in 1993. It will strengthen the research functions of the seven regional Government Industrial Research Institutes (GIRIs) under AIST's umbrella. The new policies are designed to correct for an overconcentration of industry.

Promoting regional factory siting alone will not invigorate the regions. Unique R&D bases must also be upgraded. AIST aims to make particular use of the regional research resources centering on the regional GIRIs. It will also marshal public testing and research institutes (*kohsetsushis*), third-sector research organs, and private companies. Together, these forces will play a guiding role in the regional development of technology and personnel.

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 195 Medium 196 High 197

Micro Review

continued from p. 6

he shunned the friend who had outscored him and ran home crying to his mother. In high school he discovered the representation of sines and cosines in terms of complex exponentials. When he found out that Euler had discovered the same result 150 years earlier, he was so mortified that he hid the papers on which he had written his work. At Cambridge, he prepared dinner in his rooms for some Indian friends. When one of them politely declined a third helping of his soup, he was so upset that he slipped out the back door and was not heard from for days.

Those eager to show the relevance of Ramanujan's work to practical problems have pointed out applications of his theorems to particle physics, statistical mechanics, computer science, and cryptology. These, however, were not the motivation for his work. Ramanujan was a pure mathematician whose overriding desire was for the leisure to pursue his researches. In the end he achieved this, but late and at great cost. Kanigel makes us think about this result and the kind of system that led to it.

Our own lives and experiences determine our reaction to biographies. This one moved me and held my attention. I hope that it affects you the same way.

1992 Personal Tax Edge—Final Filing Version (Parsons Technology, Hiawatha, Iowa; \$59)

In my first column (April 1987) I reviewed MacInTax for the Macintosh. I haven't looked at MacInTax since then (they didn't reply to my request for a review copy this year), but the Parsons package for PC systems under DOS is better than the MacInTax I reviewed in 1987. It's cheaper too.

The 1992 Personal Tax Edge package organizes your data as a set of linked forms, like worksheets for a

spreadsheet program. It lets you fill them out in whatever order you like. At any entry you can open relevant supporting forms. For example, at Line 34 on your 1040 form, you can open a Schedule A and begin entering your itemized deductions. You can return to your 1040 whenever you wish, and the results from your partially complete Schedule A will appear on Line 34 and be included in your dynamically updated tax computation. You can open a "bookkeeping" window for any numeric entry and use it to enter items that comprise the entry. For example, at Line 9a of your Schedule A, you can open a bookkeeping window and list your home mortgages and the interest you paid on each. The sum will appear on Line 9a.

You can attach a text comment to any entry. Like the contents of the bookkeeping windows, these comments don't appear on the printed form, but the program keeps them as part of your permanent record.

The package has other excellent features—on-line IRS instructions for forms, an audit program to catch inconsistencies, and a what-if tool for examining the effects of options in a summary view. However, there are also a few problems with the implementation. Several times I returned to Windows after exiting from the package, only to find my Windows display partially missing or oscillating unreadably. I had to exit from Windows and restart.

I also had problems printing on my laser printer, which I normally use without difficulty from Windows. The program has two options for printing forms—plain paper versions, which are acceptable to the IRS, and laser printer versions, which are facsimiles of IRS forms. I was able to print the plain paper versions, but the print spooler reported a number of time-out errors in the process. I never did manage to print the facsimile versions.

Another complaint I have about printing is that the program prints all of your attached comments and book-

keeping windows at one per page. Since these entries are short, this is wasteful and makes them hard to keep track of.

I have an ever-changing system with interacting memory manager, screen saver, screen cache, print spooler, font manager, disk cache, and compression. I suspect that most users will not see the problems I experienced. I can't judge the quality of the accounting that went into this package, but it certainly makes an unpleasant annual task a lot easier. I recommend it.

The Computer Curmudgeon, Guy Kawasaki (Hayden, Carmel, Ind., 1992, 214 pp.; \$16.95)

A curmudgeon, according to the *Oxford English Dictionary*, is a miserly, churlish fellow. Kawasaki doesn't really qualify. His good-natured satire and Macintosh evangelism suggest a warm, passionate, approachable person.

This is a silly book, but it's fun to read. Don't look for anything profound—mostly just friendly bashing of IBM, Windows, and Apple management. For example, he suggests Bill Gates as the ideal candidate for CEO of Apple.

Kawasaki intermixes a tongue-in-cheek dictionary, excerpts from published articles, and lists like The Top Ten Computer Oxymorons. (Number one on that list is IBM innovation.) He includes a few serious pieces, like one about Robin Williams and Que's publication of a book with the same title as Ms. Williams' excellent *The Little Mac Book* (Peachpit, 1991).

Kawasaki's first rule of success in business is to under promise and over deliver. His promise for this book is simply to make you laugh. For his target audience of Macintosh fanatics, I think he succeeds in satisfying that first rule.

Fonts

Try an experiment. Sit down in a quiet place and spend a few minutes inventing desktop publishing. Think about where we were in 1982. The IBM

PC was new. Microsoft was a small player. The Macintosh was still in the future. WordStar was the ultimate in word processing. Think of the technical obstacles between that environment and today's desktop publishing.

In 1982 John Warnock and Charles Geshchke, without our advantage of hindsight, founded Adobe Systems to attack those technical obstacles. I'm sure they had to face investors and issue quarterly financial reports, but what they accomplished is a powerful argument for long-range thinking and goals.

Because these pioneers were so successful, many aspects of today's computer use seem simple, even though they're not. The next two books help us understand one of the key elements of desktop publishing—fonts.

The Windows 3.1 Font Book, David Angell and Brent Heslop (Peachpit Press, Berkeley, 1992, 184 pp.; \$12.95)

Angell and Heslop have written an extremely simple book that addresses the practical problems of choosing, managing, and using fonts in the Windows environment. Much of the pow-

erful technology is hidden, transparent. The user never needs to think about it.

The authors provide clear explanations of how to use the two main kinds of fonts available under Windows: Microsoft's TrueType fonts and Adobe's Postscript Type 1 fonts. Their philosophy is simple. They think that a clearly organized, slender book is easier to use, and just as informative as a fat one. They also think that computer books need good indexes, short chapters, and lots of charts, tips, and tricks. I agree with their philosophy, and I think they succeeded in making a book that embodies it.

Incidentally, this is just one of many fine books from Peachpit Press. You can get them to send you a catalog by calling (800) 283-9444. Other excellent titles from Peachpit are *The Little Mac Book*, *Zen and the Art of Resource Editing*, *The Macintosh Bible*, *The Dead Mac Scrolls*, and *Help! The Art of Computer Technical Support*.

The Computer Font Book, Glenn Searfoss (Osborne McGraw-Hill, Ber-

keley, 1993, 398 pp.; \$27.95)

Searfoss discusses computer fonts within the framework of the 500-year evolution of printing technology. While he has come to grips with computer science and assimilated it, he has the unmistakable air of an outsider. I find his perspective extremely interesting. It is the mirror image of the view of computer scientists like Knuth and Warnock, who have studied and assimilated the printing tradition.

Searfoss' book is deeper, more interesting, and at the same time less immediately useful than the little book by Angell and Heslop. You should read this book if you're more interested in issues than procedures.

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 183 Medium 184 High 185

IEEE COMPUTER SOCIETY PRESS

COMPUTER VISIONS

by Geoffrey de Valois

This videotape shows the many techniques of computer animation through a number of full-color presentations. It features computer-animated commercials, 3-D computer graphics, dynamic and environmental simulations, scientific modeling and visualization, physically-based modeling, and behavioral, skeletal, dynamic, and particle animation. *Computer Visions* describes the technology of rendering, digitization, referencing, texturing, plotting and testing, replication of motion, sculpture, wrapping, and movement.

RUNNING TIME: 60 MINUTES.
c.32 PAGES. SOFTCOVER HANDBOOK. ISBN 0-8186-2793-X.
VIDEOTAPE #2793-16

LIST PRICE \$49.00 — MEMBERS \$39.00

To order call: 1-800-CS-BOOKS — or — FAX 714-821-4010



New Products

Send announcements of new microcomputer and microprocessor products to
Managing Editor, IEEE Micro, PO Box 3014, Los Alamitos, CA 90720-1264.

Joe Hootman

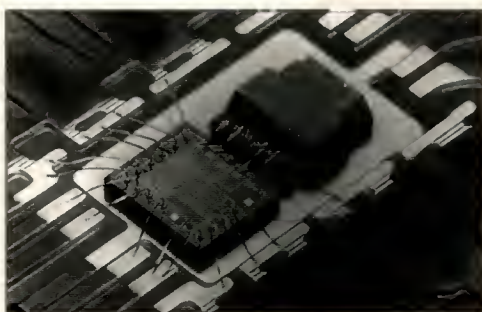
University of
North Dakota

Chips and devices

Accelerometer resists shock damage

Designed from conception for the automotive market, the MAS50G two-chip silicon accelerometer contains built-in stops to limit travel of sensing plates when disrupted by g forces. A separate, CMOS control IC inside the same package enables an MCU interface. The IC allows sensor output to be applied to a microcontrol unit used in vehicle air bag systems. Other design features include a three-layer differential capacitor, small sensing cell (size) that is hermetic at the die level, self-test capability, and plastic packaging. The single capacitive sensor combines established bulk-machining expertise for the hermetic cap and advanced surface micromachining for the capacitive sensing element. *Motorola.*

Reader Service No. 11



Motorola's MAS50G accelerometer

All-digital video synchronization

The Bt812 single-chip decoder for desktop-video capture applications uses Ultralock all-digital video synchronization technology to maintain stable incoming signals. Two internal, 8-bit, flash ADCs digitize NTSC and PAL composite or S-Video analog video signals to either digital RGB or YCrCb video data at 8-MHz to

16.5-MHz rates. Designers can adjust hue, contrast, saturation, and brightness levels via the MPU interface; color is automatically corrected. *Brooktree; \$44 (OEM quantities).*

Reader Service No. 12

ISA-compatible touch controllers

Recent versions of the company's modular touch controllers, a three-configuration, ISA-compatible chip set, share a hardware interface that makes them independent of the touch sensor. The software-based controller set consists of one IC chip and software that resides on the host system and controls communication to both host electronics and touch sensor functions. An IC and a masked 8052 processor that control the host electronics and touch sensor functions make up the hardware-based set. The serial set consists of a masked 8052 processor and communicates with the host electronics over a serial interface. *Carroll Touch; from \$10.50 (OEM quantities).*

Reader Service No. 13



Carroll Touch's touch controllers

Read/write channel count reduced

A mixed-signal design and proprietary SofTarget PRML technology let two channel devices reduce total read/write channel parts from

a typical 35 to as few as 5 components. The 100-pin QFP devices are the CL-SH3300 high-integration and the CL-SH4400 high-performance synchronous read channels. The CL-SH3300 integrates the digital and analog functions of a hard-disk drive read/write channel into an IC. The CL-SH4400 is the digital portion of a two-chip read/write channel that supports 96-MHz analog rates to and from the hard-disk assembly and provides 64-Mbps data transfer rates. *Cirrus Logic; \$25 (CL-SH3300) and \$27 (CL-SH4400); samples in 1,000s.*

Reader Service No. 14

VRAM adds graphics, multimedia video

Designers can use a 4-Mbit video RAM to add high-resolution graphics and multimedia video to workstations and high-end personal computers using Windows, X Windows, OS/2 Presentation Manager, Open Look, and Motif GUIs. Using the TMS55160 and TMS55165 VRAMs instead of DRAMs for PC graphics lets additional hardware such as shift registers be reduced or eliminated. The 70-ns and 80-ns devices incorporate JEDEC-standard features including serial register transfer, write per bit, persistent write per bit, split register transfer read, block write for fast area-fill operations, and CAS-before-RAS refresh. *Texas Instruments; \$38 (1,000s).*

Reader Service No. 15

Driver supports portables

Specifically designed for 1.3-, 1.8-, and 2.5-inch hard-disk drives used in notebook computers and other portables is the HA13526F 3.3V combination motor driver IC. The 1.7-mm HQFP device controls and drives the spindle motor that rotates the magnetic medium and the voice-coil motor that positions the read/write head. The bipolar chip includes a 1.5-A/3-phase driver and commutation logic that supports the Hall "sensorless" motor designs common in small-form-factor drives. For

safety, the HA13526F contains a built-in over-temperature shut-down circuit. *Hitachi; \$6.95 (samples, 1,000s); volume production in June.*

Reader Service No. 16

FPGAs promise 10-ns delays

A five-device family of field-programmable gate arrays enables 125-MHz operation for prescaled 24-bit counter and state machine designs, 75 MHz for 24-bit loadable counters, and 40 MHz for 24-bit accumulators. Designed for workstations, graphics, digital signal processing, and telecommunications, the ACT 3 products enable 10-ns clock-to-out delays and deliver 1,500 to 10,000 gates. With less than a 3-ns delay variance for a fan-out range of 1 to 8, designers can take advantage of the predictable timing of the devices to provide design flexibility. *Actel; from \$99.60 (100s) and \$51.10 (5,000s).*

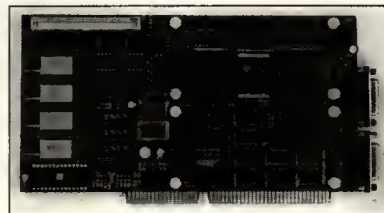
Reader Service No. 17

DSP software/hardware

C31 real-time board aids embedded market

Real-time applications board features a 40-MHz TMS320C31 digital signal processor, up to 640 Kwords of SRAM (with an option for EPROM), a dual-port memory ISA interface, and a modular approach to analog interfacing. The C31 board includes development tools such as an assembler, linker, C compiler, and a C-source level symbolic debugger. *Spectrum Signal Processing; \$2,495 each, OEM discounts available.*

Reader Service No. 18



Spectrum Signal Processing's TMS320C31 board

Processing image frames in real time

Users can extract information embedded in images acquired from visible-light cameras, tomographic equipment, radar, sonar, and infrared systems when using a board-level VMEbus system called the DSP-38. With three C30s providing 60 MIPS of power and a 68030 processor, the board directly interfaces to the MAXbus digital video interconnection system, allowing the DSP-38 to work in conjunction with other image processors. Capable of accepting either ECL- or TTL-level input clock and synchronization signals, the standard 6U double-width VMEbus module receives 32-bit-wide pixel data at 10- and 20-MHz rates. The DSP-38 requires a 5V, 20-amp power supply. *PC/M Corp.; delivery 30 days.*

Reader Service No. 19

D/A converter supports digital audio

A delta-sigma digital-to-analog converter implements 8X interpolation and 64X oversampled modulation for a low-noise DC-to-20-kHz audio bandwidth. The 28-pin plastic DIP or SOIC CS4303 supports outboard D/A processors and studio applications with greater than 100-dB signal-to-noise+distortion performance. A flexible, four-mode serial interface accepts two channels of 16- or 18-bit audio data. *Crystal Semiconductor; \$29 (1,000s); available from stock.*

Reader Service No. 20

Besting Bessel filters

According to the manufacturer, its three low-pass, 14-pin DIP or 16-lead SOIC filters with phase compensation will produce faster roll-off than classical Bessel filters while maintaining linear phase. The maximum cut-off frequency of the LTC1064-7 is 100 kHz, that of the low-power LTC1164-7 is 20-kHz, and that of the high-speed LTC1264-7 is 250 kHz. Each clock-tunable device promises a transient response that exhibits less than 5 percent over-

shoot with no ringing and operates from a +5V power supply. *Linear Technology*; from \$11.25 (100s)

Reader Service No. 21

C code generator

Users cannot only simulate and test algorithms but also generate C code to run the algorithms outside the program when using a recently introduced C code generator. Version 1.0 of the generator works with visually programmed Hypersignal-Windows Block Diagram, an object-oriented simulation program for signal processing.

To implement a DSP design, Block Diagram proves the algorithm, debugs it, and sets up What-If situations in a visual environment. The code generator then produces C code for the algorithm so the program can run in real time on a DSP. *Hyperception*; \$2,995 (code generator), \$1,995 (Block Diagram).

Reader Service No. 22

Display devices

Read portable display in sunlight

Designed for use in portable and battery-powered applications in medical equipment, industrial controllers, and test and measurement instruments is the EL240.64 full-graphic display. Wide-angle (160°) viewing, 200-fl brightness, high contrast, crisp definition, and fast screen response capabilities allow it to be viewed in outdoor applications. In normal mode with a typical screen of text and graphics, the display consumes 1.2W of power; in sunlight-readable mode it consumes under 4W. The 240x64-pixel EL display may be driven at 60-Hz to 240-Hz frame rates to vary the brightness. *Planar Systems*; \$335 (100s).

Reader Service No. 23

Producing color 3D prints of images

A 3D imaging technology called Multi-Mode Stereoscopic Imaging, or MMSI, allows users to switch between multiple viewing modes and produce color hard-

copy stereo images from printers. According to the company, MMSI can be used with most display devices and systems, such as television, and produces 20x30-inch hard copies on laser printers and offset printing systems.

Users see stereoscopic images through polarized spectacles in the binocular viewing mode; they can switch to auto-stereo mode for a more-restricted viewing mode without the spectacles. A third mode, the 3D/2D mode, lets users switch between viewing in 3D and viewing either the left or right 2D images.

MMSI uses a plastic sheet containing a checkerboard-like array of polarizing elements as small as 20 μ m. This μ Pol micropolarizer array is fabricated by a patented process so that alternating squares in the checkerboard pattern have polarization directions that differ by 90 degrees. Capable of use with most projection systems, MMSI can be licensed for use in medicine, entertainment, military, space, robotics, electronics, and science applications. *Reveo*.

Reader Service No. 24

Thin panel supports portables

A PC-compatible CPU with integrated LCD graphic display in a 3.8x6.0x2.25-inch form requires +5V power to operate, making it well-suited to portable applications. The FPMX with 1-Mbyte system RAM and 512-Kbyte flash disk capacity provides a 320x240-resolution monochrome, CGA-compatible, backlit or reflective LCD that supports 16-level gray-scale text and 4-level gray-scale graphics. An external light sensor input allows software-controlled backlight intensity to be automatically adjusted to compensate for variations in ambient light. *Mesa Electronics*; \$549 (25s).

Reader Service No. 25

Super VGA touch monitor

VARs wanting to add touch screen input to their application can use a plug-and-play color product called TM-2. The TruePoint TM-2 touch monitor with 14-inch Super VGA screen supports pixel-

by-pixel control for merchandising, information kiosk, multimedia, business, and training applications. Features include mouse-emulation drivers for DOS, Windows, and OS/2 and a resolution of 1,024x1,024 touch points. *Micro-Touch Systems*; \$1,599 including video and serial cables, drivers, software tools, and demo software.

Reader Service No. 26



MicroTouch Systems' TM-2

Remotely adjust grouped monitors

Users can remotely adjust a group of monitors such as video walls or elevated flight information monitors when using the DVG series of VGA color monitors. The 27-inch diagonal series uses Handheld Addressable Remote Control (HARC) technology for difficult-to-adjust monitor groupings. In addition to standard adjustments, each individual monitor receives a factory preset address that can be reset by a selector switch on the rear of the monitor. *Dotronix*.

Reader Service No. 27

Design software

DOS, Windows, OS/2 Verilog simulator

Design engineers using a PC at home to develop and debug Verilog HDL models may want to license a recently introduced simulator and then transfer designs to PC or workstation-based work environments. The VeriWell/PC

simulator runs in DOS, Windows, and OS/2 environments to support many behavioral, RTL, and synthesis constructs. *Wellspring Solutions; \$499.*

Reader Service No. 28

PCI simulator enhanced

Intel's original PCI Bus interface model in VHDL has been enhanced, in agreement with Intel, with a simulation model that lets users also write Verilog source commands to control operation. Commands supported by the simulator include interrupt acknowledge, I/O read and write, memory read and write, configuration read and write, memory write and invalidate, memory ready long, and postable memory write. *Logic Modeling Corporation; \$15,000 (perpetual site license).*

Reader Service No. 29

SI interfaces aid testing

With signal integrity interfaces to Cadence and Mentor Graphics layout tools, users can extract information from the layout vendor's netlist and part geometry databases for simulation testing before, during, and after layout development. Designers can highlight the nets to be analyzed and select the desired coupling distances and levels. SI's RLCG field solver generates R, L, G, and C matrices, as well as S parameters, directly from layout information. Its underlying SPICE engine solves difficult, complex problems. Interfaces are available for Allegro, Board Station, Board Station 500, MCM Station, and Hybrid Station. *CAE Division, Contec Microelectronics USA; from \$19,000 (Sun or HP workstation versions).*

Reader Service No. 30

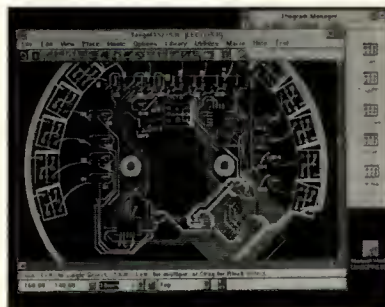
PC-compatible EDA suite

PCB designers can use the TangoPRO series of IBM PC-compatible electronic design automation software to address digital and analog design requirements. The PCB design tools with 32-bit architecture run under Microsoft Windows 3.1 and offer database, copper pour, curved traces,

one-tenth-degree entity rotation, and metric support features. Designed with object-oriented programming techniques, the TangoPRO suite consists of PCB, Route, and Schematic packages.

TangoPRO PCB lets designers define every attribute on the board and supports 99 layers, 88 of which may be defined without limit to the number of copper planes. The program's design methodology allows netlist information to be maintained with on-line electrical rules checking. *Accel Technologies; \$5,950 (PCB), upgrade discounts available.*

Reader Service No. 31



Accel Technologies' TangoPRO PCB

Math package enhances CombiScopes

An optional Math+ package for the Philips PM 338X and PM 339X family of CombiScopes helps engineers obtain signal information. Math+ performs waveform processing, including differentiation to find slew rates, integration to determine power dissipation values, and FFTs for frequency domain analysis. Histogram functions support statistical analysis, pass/fail testing using waveform templates, built-in envelope generation, and automatic limit testing on V_{PP} , V_{RMS} , frequency, and other measurement parameters. *John Fluke Mfg. Co.; \$500 (Math+), from \$5,000 (CombiScopes).*

Reader Service No. 32

Simulator features time step control

Analyzing digital IC designs as large as 100,000 transistors should be faster

and more accurate with an enhanced version of HSPICE, the analog circuit simulator. HSPICE H92A's speed improvement results from a new time step control algorithm that automatically adjusts the circuit time step size to the circuit's activity, speeding up simulation runtimes. The new models being offered with H92A work with the company's existing installed models. H92A runs on Sun, DEC, IBM, HP, Apollo, and Silicon Graphics workstations, Crays, VAXes, and 386/486 PCs. *Meta-Software; \$4,000 to \$120,000, depending on platform.*

Reader Service No. 33

Communication devices, software

Card aids PC communications

Offering 8- to 64-track communications is a 386SX/486SLC-based plug-in card for personal computers that transfers data at 4 Mbytes/s in synchronous mode and 250 Kbps asynchronously. Users can load the MCX card with the MS-DOS operating system and work on the card as if on the host PC; a driver shares resources between the card and host. Key combinations allow users to move from the host screen to a virtual monitor linked to the MS-DOS session of the MCX. In addition, a connector can link two PC-format extension cards. *ACKSYS, GD California or French Technology Press Office, Chicago.*

Reader Service No. 34

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 189 Medium 190 High 191

Product Summary

Joe Hootman

University of North Dakota

Manufacturer	Model	Comments	R.S.#
Chips			
Associated Components Technology	AIC 1812 inductors	Surface-mount series designed for LANs and peripherals is shielded to prevent coupling. The chip inductors offer 0.15 to 38.0 DC resistances, 30 to 60 Qs, and inductance values from 0.10 μ H to 1,000 μ H. <i>21 cents each (10,000s), 18 cents (100,000s); six weeks ARO.</i>	80
Calex	5D12.x DC/DC converters	Four-model series operates at 5-mV _{pp} noise in -40°C to 100°C temperatures. A shielded, low-interwinding capacitance transformer keeps output noise and input-to-output capacitance low. <i>\$48.67 each (OEMs).</i>	81
Boards			
Adaptec	AHA-1540C adapters	Enhanced 16-bit ISA-to-SCSI host adapter family can be configured by software controlled through the keyboard to eliminate physical setup of jumpers and resistors. To aid installation, the AHA-1540C's configuration appears on the computer screen. Once the stand-alone adapter is installed, SCSI peripherals can be added without opening the PC chassis. <i>From \$289.</i>	82
Anallogic	LS/MS/HS series	Data acquisition series now includes DriverLinx, a real-time Windows driver for C and Visual Basic. The multitasking, multi-user interface to PC boards ports data collection, instrumentation, monitoring, and control applications into Windows environments. Requires Windows 3.1, DOS 3.1 or higher, and C or Visual Basic. <i>\$400.</i>	83
DPT	SmartCache Plus SCSI controller	Option for Unix, multimedia, and networking users of Dell turnkey systems provides SCSI connectivity and features a design that permits users to scale SCSI performance to meet expanding needs. SmartCache Plus can be configured as a noncaching and caching controller.	84
Philips Semiconductors	OM4280 evaluation board	Controlled via a PC's RS-232 port, this kit makes use of an on-chip encryption/decryption accelerator to let users evaluate the company's smart-card microcontroller. Includes a card reader, two P83C852-based preprogrammed smart cards with secret and public keys, demonstration software, and documentation. <i>DM750, with power supplies and interconnection cabling.</i>	85
SBE	VCOM-47 FDDI controller	Single-slot, 6U VME controller with MC68839 FDDI system interface provides fiber optic LAN with P2 connectivity. Includes 25-MHz 68EC040 for on-board protocol processing, a 2-Mbyte EPROM or 1-Mbyte flash memory, and 4-Mbyte multiported DRAM. <i>\$4,995 (100s), 90 days ARO.</i>	86

FOR DISPLAY ADVERTISING INFORMATION, CONTACT:

Southern California and Mountain States: Richard C. Faust, Douglas C. Faust, 24050 Madison Street, Suite 101, Torrance, California 90505; Tel: (310) 373-9604; Fax: (310) 373-8760.

Northern California and Pacific NW: William W. Hague, 9017 Peacock Hill Road, Gig Harbor, Washington 98332; Tel: (206) 858-7575; Fax (206) 858-7576.

East Coast: Gail A. Frank, Nancy Insera, 82 Bethany Road, Suite 4, Hazlet, New Jersey 07730; Tel: (908) 264-1100; Fax: (908) 264-4340.

New England: Paul Gillespie, PO Box 6444, Holliston, Massachusetts 01746; Tel: (508) 429-8907; Fax (508) 429-8684.

Southwest: Frank E. Johnson, 3601 Smith-Barry Road, Suite 103, Arlington, Texas 76013; Tel: (817) 275-2651; Metro Voice/Fax: (817) 265-3811.

Advertising Manager: Heidi Rex, 10662 Los Vaqueros Circle, Los Alamitos, California 90720-1264; Tel: (714) 821-8380; Fax: (714) 821-4010.

For production information, conference, and classified advertising, contact Marian Tibayan. *IEEE MICRO*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, California 90720-1264; Tel: (714) 821-8380; Fax: (714) 821-4010; email: m.tibayan@compmail.com.

Moving?

**PLEASE NOTIFY
US FOUR WEEKS
IN ADVANCE**

Name (Please Print)

New Address

City

State/Country

Zip

Mail to:
IEEE Computer Society
Circulation Department
PO Box 3014
10662 Los Vaqueros Circle
Los Alamitos, CA 90720-1264

**ATTACH
LABEL
HERE**

- This notice of address change will apply to all IEEE publications to which you subscribe.
- List new address above.
- If you have a question about your subscription, place label here and clip this form to your letter.

RS # Page #

Accel Technologies	31	86
Actel	17	84
Adeptec	82	87
Analogic	83	87
Associated Components Technology	80	87
Brooktree	12	83
Calex	81	87
Carroll Touch	13	83
Cirrus Logic	14	84
Contec Microelectronics USA	30	86
Crystal Semiconductor	20	84
Dotronix	27	85
DPT	84	87
French Technology Press Office	34	86
GD California	34	86
Hitachi	16	84
Hyperception	22	85
John Fluke Mfg. Co.	32	86
Linear Technology	21	85
Logic Modeling Corp.	29	86
Mesa Electronics	25	85
Meta-Software	33	86
MicroTouch Systems	26	85
Motorola	11	83
PC/M Corp.	19	84
Philips Semiconductors	85	87
Planar Systems	23	85
Raveo	24	85
SBE	86	87
Spectrum Signal Processing	18	84
Texas Instruments	15	84
Wellspring Solutions	28	86

THE FOLLOWING INFORMATION IS AVAILABLE:

Contact the Publications Office; to facilitate handling, please request by number.

- Membership application, student #203, others #202
- Periodicals subscription form for individuals #200
- Periodicals subscription form for organizations #199
- Publications catalog #201
- Comppmail electronic mail brochure #194
- Technical committee list/application #197
- Chapters lists, start-up procedures #193
- Student scholarship information #192
- Volunteer leaders/staff directory #196
- IEEE senior member grade application #204 (requires ten years practice and significant performance in five of those ten)

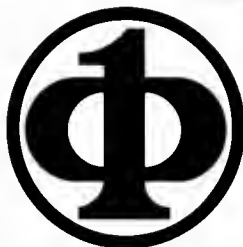
To check membership status or report a change of address, call the IEEE toll-free number, 1-800-678-4333. Direct all other Computer Society-related questions to the Publications Office.

PURPOSE

The IEEE Computer Society advances the theory and practice of computer science and engineering, promotes the exchange of technical information among 100,000 members worldwide, and provides a wide range of services to members and nonmembers.

MEMBERSHIP

Members receive the acclaimed monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.



IEEE COMPUTER SOCIETY®

A member society of the
Institute of Electrical and Electronics Engineers, Inc.

PUBLICATIONS AND ACTIVITIES

Computer. An authoritative, easy-to-read magazine containing tutorial and in-depth articles on topics across the computer field, plus news, conferences, calendar, interviews, and product reviews.

Periodicals. The society publishes eight magazines and seven research transactions. Refer to membership application or request information as noted at left.

Conference Proceedings, Tutorial Texts, Standards Documents. The Computer Society Press publishes more than 100 titles every year.

Standards Working Groups. More than 100 of these groups produce IEEE standards used throughout the industrial world.

Technical Committees. More than 30 TCs publish newsletters, provide interaction with peers in specialty areas, and directly influence standards, conferences, and education.

Conferences/Education. The society holds about 100 conferences each year and sponsors many educational activities, including computing science accreditation.

Chapters. Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

OMBUDSMAN

Members experiencing problems — magazine delivery, membership status, or unresolved complaints — may write to the ombudsman at the Publications Office.

EXECUTIVE COMMITTEE

President: James H. Aylor*
University of Virginia
Thornton Hall/Electrical Engineering
Charlottesville, VA 22903
Phone: (804) 924-6100
Fax: (804) 924-8818
E-mail: jha@virginia.edu

President-Elect: Laurel V. Kaleda*
Past President: Bruce D. Shriver*

VP, Technical Activities: Joseph Boykin (1st VP)*
VP, Conferences and Tutorials: Anneliese Von Mayrhauser (2nd VP)*
VP, Educational Activities: Gerald L. Engel†
VP, Membership Activities: Fiorenza C. Albert-Howard*
VP, Press Activities: Ronald G. Hoelzeman†
VP, Publications: Barry W. Johnson†
VP, Standards Activities: Gary S. Robinson†

Secretary: Mario R. Barbacci*
Treasurer: Michael Evangelist†
IEEE Division V Director: Bill D. Carroll†
IEEE Division VIII Director: V. Tom Rhyne†
Executive Director: T. Michael Elliott†

*voting member of the Board of Governors
†nonvoting member of the Board of Governors

BOARD OF GOVERNORS

Term Expiring 1993:

Fiorenza C. Albert-Howard, Jon T. Butler,
Michael C. Mulder, Yale N. Patt, Benjamin W. Wah,
Ronald Waxman, Akihiko Yamada

Term Expiring 1994:

Mario R. Barbacci, L. Felipe Cabrera, Wolfgang K. Giloi, Guylaine M.
Pollock, John P. Riganati, Ronald D. Williams, Thomas W. Williams

Term Expiring 1995:

Fletcher J. Buckley, Doris L. Carver, Elliot J. Chikofsky,
Joanne E. DeGroat, Michael J. Flynn,
Mary Jane Irwin, Grace C.N. Wei

Next Board Meeting

May 21, 1993, 8:30 a.m.
Stouffer Harborplace Hotel, Baltimore, MD

SENIOR STAFF

Executive Director: T. Michael Elliott
Publisher: H. True Seaborn
Director, Conferences and Tutorials: Anne Marie Kelly
Director, Finance and Information Services: Deborah Rafal
Director, Board and Administrative Services: Violet S. Doan
Assistant to the Executive Director: Sandra K. Pfau

COMPUTER SOCIETY OFFICES

Headquarters Office

1730 Massachusetts Ave. NW
Washington, DC 20036-1903
Phone: (202) 371-0101
Fax: (202) 728-9614
E-mail: hq.ofc@compmail.com

Publications Office

10662 Los Vaqueros Cir.
PO Box 3014
Los Alamitos, CA 90720-1264
Membership and General Information:
(714) 8218380
membership@compmail.com
Publication Orders: (800) 272-6657
Fax: (714) 821-4010
E-mail: cs.books@compmail.com

European Office

13, Ave. de L'Aquilon
B-1200 Brussels, Belgium
Phone: 32 (2) 770-21-98
Fax: 32 (2) 770-85-05
E-mail: euro.ofc@compmail.com

Asia/Pacific Office

Ooshima Building
2-19-1 Minami-Aoyama, Minato-ku
Tokyo 107, Japan
Phone: 81 (3) 3408-3118
Fax: 81 (3) 3408-3553
E-mail: tokyo.ofc@compmail.com



IEEE OFFICERS

President: Martha Sloan
President-Elect: H. Troy Nagle
Past President: Merrill W. Buckley, Jr.

Secretary: Souguil J.M. Ann
Treasurer: Theodore W. Hissey, Jr.

VP, Educational Activities: Edward A. Parrish
VP, Professional Activities: Charles K. Alexander
VP, Publication Activities: Helen M. Wood

VP, Regional Activities: Luis T. Gandia
VP, Standards Activities: Wallace S. Read
VP, Technical Activities: Donald M. Bolle

Sweetening the Pot!



**Hedge your
bet for the
biggest
game
around.
Subscribe to
IEEE Micro.**

**We promise
to keep you
in the chips
on the
technical
issues that
most
interest
you.**

Be sure to read *IEEE Micro*'s next issue. June features articles from the IEEE Computer Society TCM's Hot Chips IV Symposium:

- Performance Features of the PA7100 Microprocessor—Hewlett Packard
- The Alpha AXP Architecture and 21064 Processor—DEC
- Architecture of the Pentium—Intel
- Sparcle: An Evolutionary Processor Design for Multiprocessors—MIT

Keep on reading and you'll find more of the information you need in columns specifically tailored to your needs by industry professionals:

- Software/hardware reviews—Richard Mateosian
- Comments on standards developments—Steve Diamond
- Legal matters affecting your products—Richard Stern
- News from the Pacific Rim—David Kahaner
- Interviews with key industry people—Ware Myers
- Industry news, new products, and more

IEEE MICRO

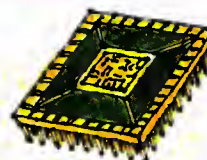
Keep current with IEEE Micro!

IEEE

MICRO

JUNE 1993

Chips, Systems, Software, and Applications



HOT CHIPS IV

Serving up a new batch



In this issue:

- ☐ Hewlett-Packard PA7100
- ☐ DEC Alpha AXP and 21064
- ☐ Intel Pentium
- ☐ MIT Sparcle Multiprocessor

 IEEE COMPUTER SOCIETY

 THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.

IEEE MICRO

Published by the IEEE Computer Society

Volume 13 Number 3

June 1993

F E A T U R E S

THEME ARTICLES

**9 Guest Editor's Introduction:
"Hot" and "Cool" Chips**

John R. Masbey

**11 Architecture of the Pentium
Microprocessor**

Donald Alpert and Dror Avnon

Maximizing performance while preserving software compatibility within the practical constraints of technology

**22 Performance Features of the PA7100
Microprocessor**

Tom Asprey, Gregory S. Averill, Eric DeLano, Russ Mason, Bill Weiner, and Jeff Yetter

Combining an integer core and floating-point coprocessor into a single-chip format

**36 The Alpha AXP Architecture and 21064
Processor**

Edward McLellan

Featuring 64-bit, 200-MHz operation plus initial ports for Unix (DEC OSF/1), OpenVMS, and Windows NT operating systems

**48 Sparcle: An Evolutionary Processor
Design for Large-Scale Multiprocessors**

Anant Agarwal, John Kubiawicz, David Kranz, Beng-Hong Lim, Donald Yeung, Godfrey D'Souza, and Mike Parkin

Providing fast message handling, latency tolerance, and fine-grain synchronization through minor modification to an existing microprocessor

SPECIAL FEATURES

**62 Message-Routing Systems for
Transputer-Based Multicomputers**

Domenico Talia

Efficiently exchanging data among processes mapped on transputers that are not directly connected



Cover: Jay Simpson, Design and Direction

Circulation: *IEEE Micro* (ISSN 0272-1732) is published bimonthly by the IEEE Computer Society, PO Box 3014, Los Alamitos, CA 90720-1264; IEEE Computer Society Headquarters, 1730 Massachusetts Ave., NW, Washington, DC 20036-1903; IEEE Headquarters, 345 East 47th St., New York, NY 10017. Annual subscription: \$23 in addition to IEEE Computer Society or any other IEEE society member dues; \$42 for members of other technical organizations. This journal is also available in microfiche form.

Postmaster: Send address changes and undelivered copies to *IEEE Micro*, PO Box 3014, Los Alamitos, CA 90720-1264. Second-class postage is paid at New York, NY, and at additional mailing offices. Canadian GST#125634188.

Copyright and reprint permissions: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US Copyright Law for private use of patrons those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 29 Congress St., Salem, MA 01970. For other copying, reprint, or republication permission, write to Permissions Editor, *IEEE Micro*, PO Box 3014, Los Alamitos, CA 90720-1264. Copyright © 1993 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved.

Printed in USA.

DEPARTMENTS

- 4 On the Edge**
Desktop document management
- 7 Micro News**
Design services; micro applications; manufacturing fellowships; PDAs
- 73 Micro View**
FPGA update
- 75 Micro Law**
Glitches in software copyrights
- 81 Micro Standards**
Organizing corporate standards
- 84 Micro Review**
Manuals and guest reviews
- 88 Software Report**
Assembly line automation
- 93 New Products**
CAD tools; DSP systems; communications/displays
- 98 Product Summary**
- 100 Author Guidelines**

*Reader Service cards, p. 96A/B;
Advertiser/Product Index/
Moving Coupon, p. 104;
Computer Society information,
cover 3*

IEEE Computer Society
PO Box 3014, Los Alamitos, CA 90720-1264
(714) 821-8380

Editorial: Unless otherwise stated, bylined articles and descriptions of products and services reflect the author's or firm's opinion; inclusion in this publication does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society. Send editorial correspondence to *IEEE Micro*, PO Box 3014, Los Alamitos, CA 90720-1264. All submissions are subject to editing for style, clarity, and space considerations.

EDITOR IN CHIEF

Dante Del Corso
*Politecnico di Torino**

ASSOCIATE EDITORS IN CHIEF

K.E. Grosspietsch
GMD

Ashis Khan
Mips Technologies, Inc.

Maurice Yunik
University of Manitoba**

EDITORIAL BOARD

Stephen L. Diamond
SmsSoft, Inc.
Joe Hootman
University of North Dakota
Victor K.L. Huang
National University of Singapore
David K. Kahaner
National Institute of Standards and Technology
Hubert D. Kirmann
Asea Brown Boveri Research Center
Priscilla Lu
AT&T
Richard Mateosian
Teresa H. Meng
Stanford University
Nadine E. Miner
Sandia National Laboratories

Gilles Privat
France Telecom
Ken Sakamura
University of Tokyo
John L. Schmalzel
University of Texas at San Antonio
Arun K. Sood
George Mason University
John W. Steadman
University of Wyoming
Richard H. Stern
Oblon, Spivak, McClelland, Maner & Nenstadt
Osamu Tomisawa
Mitsubishi Electric Corporation
Philip Treleaven
University College London
Uri Weiser
Intel Israel, Ltd.

MAGAZINE OPERATIONS COMMITTEE

James J. Farrell, III (chair)
Valdis Berzins
B. Chandrasekaran
Carl Chang
Manuel d'Abreu
Dante Del Corso
John A.N. Lee
Ted Lewis
Michael J. Quinn
Peter R. Wilson

PUBLICATIONS BOARD

Barry W. Johnson (chair)
James J. Farrell, III
Ronald G. Hoelzeman
Mary Jane Irwin
Ming T. (Mike) Liu
Michael C. Mulder
Theo Pavlidis
David C. Rine
Sallie V. Sheppard
Harold Stone
Kishor Trivedi

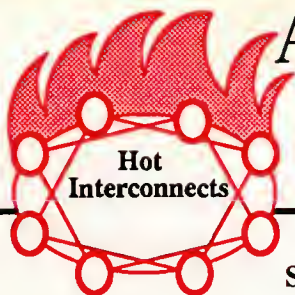
STAFF

Marie English
Managing Editor
Dick Price
Staff Editor
H.T. Seaborn
Publisher
Marilyn Potes
Editorial Director, CS Magazines
Jay Simpson
Art Director
Joseph Daigle
Design/Production
John Gill
Membership/Circulation Manager
Heidi Rex
Advertising Manager
Marian Tibayan
Advertising Coordinator

* Submit six copies of all articles and special-issue proposals to Dante Del Corso, Dipartimento di Elettronica, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129 Torino, Italy; phone +39-11-564-4044; Compmail: d.delcorso; Internet: delcorso@polito.it; Bitnet: delcorso@itopoli or

** Maurice Yunik, Dept. of Electrical Engineering, University of Manitoba, Winnipeg, Manitoba R3T 2N2, Canada; phone (204) 474-8517; yunik@eeserv.ee.umanitoba.ca

Author guidelines for article submission are available from the West Coast office (address at left).



Advance Program

Hot Interconnects'93

A Symposium on High Performance Interconnects

Sponsored by TCMM, the Technical Committee on Microprocessors and Microcomputers of the IEEE Computer Society



Stanford University, Palo Alto, California -- August 5-7, 1993

General Chair: Hasan AlKhatib, Santa Clara University **Program Co-Chairs:** Paul Borrill, Sun Microsystems, and Anoop Gupta, Stanford University

Hot Interconnects is a symposium on high performance buses, multiprocessor interconnects, networks, and interfaces. Like Hot Chips, it is intended for the engineering practitioner who is interested in real solutions. Hot Interconnects is a platform for sharing knowledge and solutions among the communities of architects and designers of single systems, I/O subsystems, and networks of systems.

August 5, 1993 — Kresge Auditorium

- 8:45 – 9:00 **Welcome and Opening Remarks**
Hasan AlKhatib, General Chair
Paul Borrill and Anoop Gupta, Program Co-Chairs
- 9:00 – 10:00 **Keynote Address: Ed McCracken, President and CEO, Silicon Graphics Inc. "Interconnection Issues at Every Level in the Computer Industry: Opinions & Futures"**
- 10:30 – 12:00 **Advanced Bus Technology**
Session Chair: Paul Borrill, Sun Microsystems
• "The Challenge Interconnect: Design of a 1.2 GByte per sec Coherent Multiprocessor Bus" Mike Galles, SGI
• "The HP suMMit Bus (PMB)" Syrus Ziai and Cheryl Ranson, HP
• "Electrical Design of the XDBus Using Low Voltage Swing CMOS (GTL) in the SparcCenter 2000 Server" Christopher Cheng and Leo Yuan, Sun Microsystems
• "LTL – A High Performance CMOS Transceiver Technology" Bob Lipp, Consultant
- 12:30 – 2:00 **Lunch**
- 2:00 – 4:00 **MP Interconnects**
Session Chair: Anoop Gupta, Stanford University
• "The Meiko CS2 Interconnect" Mark Homewood, Meiko
• "The SCI NodeChip, a 500 MByte per sec Virtual Backplane Chip" Hakon Bugge and Knut Alnes, Dolphin SCI Technology
• "The S3.mp Interconnect System and TIC Chip" Andreas Nowatzky and Mike Parkin, Sun Microsystems
• "A Pipelined Network Interface for a Parallel Computer" Gunter Watzlawik and Franz Hunter, Siemens
- 4:15 – 5:45 **Interconnect Design and Analysis**
Session Chair: George Cox, Intel
• "A Cost and Speed Model for k-ary n-cube Wormhole Routers" Andrew Chien, Univ of Illinois, Urbana-Champaign
• "Hnet: A High-Performance Network Evaluation Testbed – Preliminary Results" David Smitley, Frank Hady, and Dan Burns, Supercomputing Research Center
• "Characterizing the Performance Space of Shared-Memory Machines Using Micro-Benchmarks" Rafael Saaverda, R. Stockton Gaines, and Michael Carlton, USC

5:45 – 7:15 Reception

7:15 – 9:00 **Thursday Evening Panel: Benchmarking Interconnects**
Moderator: George Cox, Intel
• Panel Members TBD

August 6, 1993 — Kresge Auditorium

- 8:45 – 10:00 **Keynote Debate Moderated by Gordon Bell "State of the ATM Art"**
Larry Roberts, NetExpress
"Is ATM the only Answer to All Networks?"
Danny Cohen, ISI
- 10:30 – 12:30 **High Speed Networks**
Session Chair: Van Jacobson, LBL
• "The VIC (Virtual Interconnect with Control) Proposal for Maximizing ATM Networks Performance" H.T. Kung, Harvard University
• "ATM over Fibre Channel: the ANSI Solution" Terry Anderson, Ancor
• "Atomic: From Interconnection Network to Local Area Network" Bob Felderman, USC/ISI
• "Multiplexing on a High Speed Network" Bryan Cook, IBM Poughkeepsie
- 12:30 – 2:00 **Lunch**
- 2:00 – 3:00 **Gigabit Links**
Session Chair: Greg Chesson, SGI
• "The HP 2-Gbit per sec G-Link Chipset" Chu Yen, HP
• "Integration of Multiple Bidirectional Point to Point Serial Links in the Gbits per sec Range" Roland Marbot, Bull S.A.
- 3:15 – 4:15 **New Developments**
Session Chair: Deborah Estrin, USC
• "Hybrid Access Systems to the Internet" Howard Strackman, Hybrid Networks
• "New Technology in the IEEE P1394 Serial Bus – Making it Fast, Cheap and Easy to Use" Michael Teener, Apple Computer
- 4:30 – 6:30 **Wireless Technologies**
Session Chair: Kathleen Nichols, Apple Computer
• "New Markets, a New FCC, and New Spectrum" Jim Lovette, Apple Computer
• "Key Issues in Wireless Networking" Chih-Lin I, AT&T Bell Labs
• "CDMA Data Services" Phil Karn, Qualcomm
• "Mobile Internetworking" Steve Deering, Xerox PARC

Organizing Committee

Publicity: Kathleen Nichols, Apple Computer, and
Glen Langdon, University of California, Santa Cruz

Treasurer: Dima Khoury, TTC of Silicon Valley

Digest: Nam Ling, Santa Clara University

Registration: Qiang Li, Santa Clara University

Local Arrangements: Robert Stewart, Stewart Research

Program Committee:

Paul Borrill, Sun Microsystems
Anoop Gupta, Stanford University

Hasan AlKhatib, SCU

Dal Allen, ENDL

Gordon Bell

David Cheriton, Stanford University

Greg Chesson, Silicon Graphics

George Cox, Intel

Deborah Estrin, USC

Tom Knight, MIT

Kathleen Nichols, Apple Computer

Tutorials, Saturday, August 7, 1993.

8:30 - 12:00

Tutorial #1:

Part I: "Opto Electronics" by
Schelto van Doorn (Siemens),
Ron Soderstrom (IBM), and
Gerry Rawls (Finisar)

Part II: "Fibre Channel" by Todd
Sprenkle, Tandem

8:30 - 12:00

Tutorial #2:

"Multiprocessor Interconnects"
by Andrew Chien, University of
Illinois, Urbana-Champaign
Lunch

12:00 - 1:30

1:30 - 5:00

Tutorial #3:

"ATM" by Allan Fisher and Peter
Steenkiste, CMU

Registration includes: attendance; two lunches; parking; one copy of notes; coffee breaks; and Thursday evening reception. On-site registration will start at 7:30 am on Thursday, August 6, 93

Registration may be faxed or e-mailed if paid by credit card.

Registration Fees for Technical Program

	Post-marked by July 12, 93	After July 12, 93
IEEE-CS or ACM Member	\$200	\$250
Non-Member	\$250	\$315
Unemployed	\$180	\$230
Full-Time Student	\$100	\$120

Registration Fees per Half-day Tutorial

	Post-marked by July 12, 93	After July 12, 93
IEEE-CS or ACM Member	\$90	\$105
Non-Member	\$110	\$135
Unemployed	\$70	\$85
Full-Time Student	\$20	\$30

Hot Interconnects '93 Registration Form

Name _____

Affiliation _____

Mailing Address _____

City _____ State/Zip _____ Country _____

Telephone No. () / _____

Area Code _____ Fax No. _____

Check all that apply

☐ Please, don't include my name in any mailing list.

Membership ☐ IEEE, or ☐ ACM _____
Membership No. _____

☐ Full-time student ☐ Unemployed _____
Amount paid _____

☐ Symposium Technical Program _____

☐ Tutorial #1 or ☐ Tutorial #2 _____

☐ Tutorial #3 _____

Payment Method: (Amount enclosed: \$ _____)

☐ Check drawn on a U.S. bank, ☐ VISA, ☐ MC

Make check payable to: *Hot Interconnects Symposium*

Credit Card No. _____ Expiration Date _____

Name on Credit Card _____ Signature _____

Mail to: Dr. Qiang Li, Dept. of Computer Engr.,
Santa Clara University, Santa Clara, CA 95053

☐ Please, send me housing information.

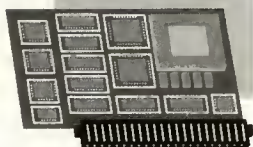
Note that Hot Interconnects and Hot Chips are scheduled back-to-back.

Thu Fri Sat Sun Mon Tue
August 5 6 7 8 9 10

Hot Interconnects Hot Chips

For more information contact Dr. Qiang Li
at (408)554-2730, Fax: (408)554-5474,
Internet: qli@sunrise.scu.edu

On the Edge



Desktop document management

[I've asked Larry Miller, vice president of advanced products and business planning at Caere Corporation, to share with us his vision of the technologies required for desktop document management. Caere is a leading supplier of optical character recognition technology. His report follows.—S.D.]

Stephen L. Diamond

SunSoft, Inc.

Phone (415) 336-4190

Fax (415) 336-4477

steve.diamond@eng.sun.com

For quite some time, information services managers and the senior executives of businesses large and small have known about the need for document management. Existing document management solutions, however, always seem to require a total conversion of a company's established processes, a whole new set of software, and most importantly, expensive hardware upgrades.

Most MIS professionals, if just to capitalize on their current hardware and software investment, have been awaiting the day when the technology will support document management in a personal computer environment. Until quite recently, such an implementation has met significant technology barriers. Exactly what those barriers are, and how they are being solved, is the topic of this column.

Compression of images

The first barrier to document management in a PC environment has to do with the documents themselves. Most documents not under control of file handling software or add-on utilities for the operating system exist outside the computing environment. They come from outside of the company, and they start out as paper. Physically getting these paper documents into a PC environment is the first challenge.

For today's desktop scanners, the most common resolution for input is 300 dots per inch (dpi). For an 8-1/2 × 11-inch piece of paper, this calls for a full megabyte of storage per page! So the typical 80-Mbyte local hard disk of a robust system today holds just 80 pages, and more likely about 40 pages with Windows software installed. Most MIS professionals have been hoping that the technology to compress these images will soon be perfected, or that storage costs will fall significantly.

What has occurred in the storage environment is that the optical WORM (write once, read many) solutions have not expanded to the typical PC environment. While WORM devices can hold a great deal of information, their access time and cost have prevented them from becoming commonplace among PC users.

Special compression chip sets, like those created by C-Cube Microsystems, once held out some hope—their real-time compression rates were outstanding. Much of the interesting work there, however, is now devoted to video, and hence shows up in specialized multimedia systems. For all the discussion at trade shows over the past few years, we have yet to come across a mainstream PC with built-in compression technology.

Faxes are basically images, and although scanned at lower resolutions (204 × 196 or 204 × 98 dpi), they present a similar problem. When imported with a fax board, a high-resolution fax requires almost a half megabyte of memory per page. We often use the CCITT Group 3 and Group 4 compression standards in imaging applications, but the net reduction is not sufficient for real PC or local area network implementation. [CCITT, or Comité Consultatif International de Télégraphique et Téléphonique,

is a Geneva-based standards-setting division of the International Telecommunications Union.—Ed.] These two methods operate by preserving the ratio of black and white dots. Group 3 does this on a line by line basis, and Group 4 does it bidirectionally. These produce pages of 180 to 250 Kbytes for Group 3 and about half that for Group 4. This is a major improvement, but will still not suffice.

We need more intelligent approaches to accomplish greater levels of compression in software. Caere Corporation of Los Gatos, California, has recently announced a new technology it has developed, called Super-Compression. This procedure first separates the pages into identified text and graphic areas. This small bit of information regarding the page provides huge benefits in later compression. [For a review of compression tools, see *Micro Review*, p. 84.—Ed.]

If you look closely at any photograph published in this magazine, you will see a pattern of dots; that very ratio of dark to light regions, in both directions, provides the information. However, in text areas, the ratio of black and white dots is irrelevant. Only the black dots have meaning. In fact one could throw out all the white dots! The benefit is clear. Just look at this page and imagine the percentage of the total surface area that is contained in the "white dots." By capturing the *X,Y* locations of the black dots, we have already taken a big step in compression.

Known as a leader in optical character recognition products, Caere has developed a considerable amount of technology for recognizing shapes. To further enhance the compression of text and graphics, this technology searches for redundant shapes and only saves the first example of each. This step allows compression ratios of up to 50:1 over the original image, a major improvement over existing technology. A page of text then can be saved in as little as 15 Kbytes, a level that existing magnetic storage can easily handle.

Using this technology, the same 80-Mbyte drive mentioned earlier could hold over 5,000 pages.

The smaller size for images also makes transmission over a typical LAN much more practical. Pages of a megabyte each, or even 250 Kbytes, make a typical three-page document prohibitive in terms of network traffic. With only 45 Kbytes needed for the same document, LAN traffic can indeed handle document management tasks.

Optical character recognition

Absolutely key to having document management come onto desktops is the state of the OCR art. Just five years ago, the best desktop OCR technology was based on a matrix matching (matching a bitmap template with a bitmap scanned from the document). This procedure is practical only for a limited number of typewriter fonts. In the real world of laser printers, multiple fonts, and magazines, such a limitation totally inhibited the use of OCR in the office. In the last five years, OCR technology has improved greatly. Omnifont OCR, or the ability to read a wide variety of fonts and font sizes, has migrated into PCs from the world of dedicated scanning machines. Caere's OmniPage was the first software-only omnifont product. Others from Xerox, Calera, and DEST soon followed.

Omnifont OCR works much differently from matrix matching OCR. Rather than compare found shapes to a predetermined collection of shapes in memory, omnifont OCR works by analyzing the shapes for various telltale characteristics. For instance, if it determines that a character has a vertical line and a dot over the top of it, the algorithm knows it is an *i*. It does not need to know how high the line is, or how thick, or exactly the space to the dot. Only the main characteristics are needed.

Simple as this sounds, omnifont OCR is extraordinarily computation-intensive. Caere's OmniPage uses full

32-bit calculations to accomplish its work and requires over a million lines of code. Before the 386 came along, it required a coprocessor card with a CPU and memory to handle the load. Now, with 386 and even 486 systems the norm, and Windows 3.1 to provide virtual memory, the computing requirements of modern omnifont OCR are easily available.

Having the technology to input and store images, together with modern OCR that can convert the text of those images into ASCII, virtually solves input and storage problems. However, a final input-type problem remains: indexing.

Traditional imaging solutions running on minicomputers and mainframes require that the user specify keywords for each image. If you are keyboarding something and have to type the few key words, or underline them with a mouse, or add a stroke of the F1 key, it's not too much to ask. However, if the document has come in automatically, and has been supercompressed and scanned without the touch of human hands, typing in the keywords is, relatively speaking, a great deal of additional work. Needed, then, is some kind of automatic indexing to ease the input burden, which so far has merely moved downstream.

Automatic indexing

Much talk at imaging trade shows lately has focused on full text indexing, or full text retrieval. This is a handy method. However, since every word (other than the typical stop words like *or*, *the*, or *a*) is indexed as important, you end up finding documents that contain words, but that may not be about the words found. An article about a beauty contest and one about a *concours d'elegance* may both contain words like *sleek*, *beautiful*, and *stunning*, but they are about very different things.

One way to weight important words and to link relevant documents is to build a thesaurus of synonyms. This is

not as simple as including a normal thesaurus, as no normal thesaurus has a linkage for IBM, International Business Machines, and Big Blue, for example. These kinds of linkages must build upon human intelligence. Programs like Verity's Topic allow users to add linkages as they use the database; they even enable programmers or system administrators to set up such webs of relevance ahead of time. As new documents come onto the file, they pass through a filter that adds values and linkages to the special words.

Caere has taken a different approach with its new product, PageKeeper. They have come up with an algorithm that identifies the topic word of each sentence. The topic word is usually the grammatical subject of the sentence. In the earlier example of a car show and a beauty contest, the adjectives and adverbs may be similar, but the subjects to which they refer are different.

This methodology has the distinct advantage of being fully automatic and astoundingly accurate in its linkages. After all, the subjects of the sentences, when taken as a whole, are what the document is about. That's the nature of language. However, therein lies the disadvantage of this approach: it is language dependent. The rules for finding the subject of a sentence vary greatly among different languages. Each language would need a special version of the algorithm written for it. Simpler indexing schemes and lists of synonyms are not inherently variable by language.

To bring document management to the desktop, we must have some kind of automatic indexing, whether preprogrammed, added with use, or fully automatic. In combination with the new OCR and compression technologies, input of documents to a PC environment can become remarkably straightforward.

Value-added retrieval

The last of these four technical challenges is to add value to the retrieval

process. If we can make the input of documents into the system as automatic as described, the barrier to entry will be low enough that a wide variety of documents will find their way into it. Imagine the variety of documents that a work group would create if all the "Please cc. (list), FYT" stickered articles and pages normally passed around by sneaker-net would just be put into the document management system. Finding such documents later and determining their relevance to the task at hand is the key.

Various approaches have been developed to accomplish this. The programmed filters will allow us to define linkages and therefore provide navigation tools through the retrieved documents. Synonym lists also provide this kind of help. But the retrieved documents come back in a list, and the document on the top of the list is not necessarily the best one. As databases grow larger, having the most relevant documents at the top becomes more important. Most readers have probably done on-line searches, but how many actually read all the retrieved material? One analyst says that in a list larger than one computer screen, most people don't read past *K*. If the best document starts with an *P*, it would never be seen.

This problem has been addressed in several ways. Some full text retrieval systems have provided a "relevance" listing that is based on the count of the found words. This helps, but long documents or bibliographies can often end up at the top without being the most relevant ones.

One company has addressed this problem by a context-based relevance weighting. Measuring the proximity of the various search words to each other in the found documents can provide a more meaningful weighting than just a rough count of found words, but it can also lead to anomalies and false positives. A car brochure mentioning an air conditioner that has passed environmental safety standards and provides quality air for the passengers

might turn up as the most relevant document in a search for environmental activities last year.

Caere uses a different, more sophisticated but language-dependent methodology to rank documents in their "weighted relevance" retrieval. The algorithm weighs documents according to the degree to which search words are used as topic words, their frequency within documents, and their relative uniqueness within the database as a whole. The results are quite impressive, even "scary" according to a well-known personal computer magazine. However, since this approach is based on language, as is our understanding of the documents, such results should not be unanticipated.

With the four main technical barriers to desktop document management either down or quickly falling, we can expect to see even more solutions in this area. Presently, solutions from Lotus, with their Lotus Notes Document Imaging for enterprise-wide installations, and Caere, with PageKeeper for work-group level adaptation, cover a wide spectrum of requirements. As groups and companies seek to improve their document handling and information sharing, such installations will become commonplace. It foretells a kind of *glasnost* between the worlds of paper and electronic communication, which, like the *glasnost* between countries, can only be good thing.

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 192 Medium 193 High 194



Send information for inclusion in Micro News one month before cover date to
Managing Editor, IEEE Micro, PO Box 3014, Los Alamitos, CA 90720-1264.

Designers: Take note

A variety of recently announced services aim to help designers get the most out of their plans. First, consider a European microelectronics service for small- and medium-size enterprises that was announced last February.

The Chip Shop. European institutes, supported by industry, help local companies using IC technology in their products reduce costs, time to market, and risks. The Esprit III-funded shop provides regular, low-cost multiproject wafer runs for prototyping and advises on design flows and the use of ASICs. Chip Shop lets the companies capitalize on the experience of trained engineers via the JESSI and CEC Special Actions networks. Chip Shop centers can be found in Germany, France, Italy, Spain, Portugal, Greece, Denmark, Norway, and the UK.

Interested parties can contact the Chip Shop Secretariat, SCME Delft, PO Box 6067, 2600 JA Delft, The Netherlands; phone +31 15 697118 or fax +31 15 571603.

Microlithography lab. FSI International announced the opening of a microlithography applications lab in Dallas, Texas. The lab focuses on process development for 200-mm wafers, including tool characterization and enhancements, but will also support product demonstrations and address customers' process issues.

Companies can access a complete Polaris Microlithography Cluster and a Sematech GCA stepper. The Polaris alternative to conventional photolithography track systems consists of a cluster of independent modules arranged around one or two Staubli Puma 560 series clean-room robots. The modules do not require a mechanical interface. This methodology, coupled with simplicity of design, allows production systems to demonstrate high MTBFs.

For more information, contact FSI International, 322 Lake Hazeltine Drive, Chaska, MN 55318-

1096; phone (612) 448-5440 or fax (612) 448-2825.

Wafer fabrication. The NCR Microelectronic Products Division will invest \$81 million in an 8-inch wafer fabrication expansion in Colorado Springs, Colorado. Plans call for volume production in the third quarter of 1994. Aimed at improving process efficiencies, NCR's facility will emphasize submicron CMOS semiconductor manufacturing technologies. Call the NCR hotline at (800) 334-5454 for more information.

IC factory. Texas Instruments plans to establish a factory designed especially to provide customers with small quantities of a variety of custom circuits rather than mass-produced ICs. Consisting of integrated object-oriented software for real-time factory and process control, distributed workstations, database server, modular process systems, and single-wafer process technology, this factory promises great flexibility in wafer fabrication.

TI's Microelectronics Manufacturing Science and Technology program will produce the smaller, million-transistor circuits in near particle-free environments in fast cycle times. New program approaches include single-wafer processing, elimination of ultra clean rooms, real-time process control, easy-to-use smart equipment, and a computer-integrated manufacturing environment.

Developed by TI under contract with US government agencies, the MMST program's object-oriented microelectronics architecture also applies to other manufacturing domains.

CIM framework. Sematech has awarded a contract to Texas Instruments to codevelop and implement an industry-standard process control software framework for computer-integrated manufacturing. The standardization goal is to produce a software applications platform that will function like Microsoft Windows on a per-

sonal computer but on a much larger multicomputer, factorywide scale.

The contract also calls for enhancement of a next-generation, TI-developed application module called Specification Manager that manages manufacturing specifications and integrates into the CIM framework. Sematech calls its framework-standard application Spec_Builder.

TI and Sematech are also discussing the possibility of standardizing and enhancing six additional applications for the framework. They would be developed as part of the MMST program.

Contact John Ahearne, MS8434, Texas Instruments, 6550 Chase Oaks Blvd., Plano, TX 75023, for program information.

MCM prototyping. nChip, Inc., and the US agency, DARPA, agreed to cooperate in a development program to reduce the time and cost of developing multichip module designs.

By implementing improvements in MCM package standardization, computer-automated manufacturing, design tool interfaces, and module testing, nChip expects to reduce design turnaround times and costs by factors of four. A key goal of the 2.5-year, \$5.4-million program is to reduce to less than four weeks the module development time from the hand-off of a design to nChip, to delivery of tested prototypes.

The program calls for

- a family of off-the-shelf MCM carriers to be defined and produced to eliminate tooling costs and long lead times;
- standard substrate sizes, with preprocessed power, ground, and decoupling layer, to reduce design time and allow inventorying of partially processed wafers;
- production of design kits for major EDA software tools so system designers can handle all aspects of MCM design in house; and
- a standard die library format to

identify the IC information required for the design and assembly of MCMs using bare die. Logic Modeling Corporation will lead the program's die library effort and work with industry groups to help define MCM-related standards for design tool interfaces.

nChip is headquartered at 1971 North Capitol Ave., San Jose, CA 95132; phone (408) 945-9991.

PowerOpen environment. Seven manufacturers recently announced formation of an independent corporation to support their PowerOpen association founded in 1991. PowerOpen Association, Inc., will promote the PowerOpen environment and provide software developers with services that support the development of environment-based products.

Founding members Apple, Bull, Harris, IBM, Motorola, Tadpole Technology, and Thomson-CSF support the Power PC RISC architecture, Application Binary Interface specification, and a choice of user environments.

Users will be able to plan on a single version of software that runs across many systems and reduces development costs. President Dominic J. LaCava says that the association will manage a certification process that will assist end users in assessing compatibility with the PowerOpen ABI.

Further information is available from Christine Williams in Europe, phone +44 71 637-1509; and Norm Kalat in the US, phone (508) 294-4514.

Applications galore!

Today's ubiquitous micro supports a diverse list of applications; here's a sampling to keep you up to date.

SCI products. Dolphin SCI Technology AS announces its 64-bit Node Chip, which implements the IEEE Std 1596-1992 Scalable Coherent Interface in high-performance and low-cost versions. Dolphin worked closely with the SCI working group from its beginnings as the SuperBus study group in 1987,

fabricated the chip at Vitesse Semiconductor, and plans to offer 200-/1,000-Mbyte/s-interconnection bandwidth versions.

Useful in shared-memory and message-passing massively parallel processor systems, workstation clustering, and high-bandwidth I/O interconnection, NodeChip implements a selected subset of SCI, incorporating transceivers, buffers, and transaction control. The single chip also supports cache-coherent systems based on the distributed shared-memory model of SCI.

The first products include a starter kit, an evaluation kit, and 500-Mbyte/s parts. Several companies are reported to have reserved a number of the chips to be fabricated in the first batch.

Intended primarily for developers, the chip's current prices and performance are not indicative of what can be expected in the longer term for high-volume applications. The DST501A Starter Kit with NodeChip, Cbus interface, key accessories, and design support lists at \$19,500. An evaluation kit containing NodeChip, a VME board that behaves like a simplified SCI-VME bridge, and software costs \$29,500; an additional board is \$20,000. Multiple kit discounts are available.

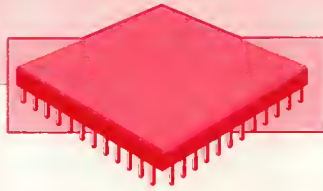
Contact Dolphin's European office via phone at +47 262 7000 or fax at +47 262 7007; scimktg@dolphin.no. The US phone and fax numbers are (603) 465-3180 and (603) 465-2680.

Wireless products. Three newly formed partnerships should ease the production of wireless products.

Communications. Sun Microsystems and two-year-old Elvis+ Ltd., a private Russian design corporation, signed a technology licensing and joint development agreement to develop wireless network communication technology. Elvis+ (short for Electronic Computer and Information Systems) will work jointly with Sun engineers on wireless communication projects and use Sun Sparcs for future workstation products.

Meter readers. Schlumberger Limited

continued on p. 102



Guest Editor's Introduction

"Hot" and "Cool" Chips

John R. Mashey

Silicon Graphics, Inc.

The annual Hot Chips Symposium is always an interesting experience. For Hot Chips IV, the program committee (chaired by Dave Patterson and myself) once again had the pleasure (and difficulty) of selecting from too many interesting papers. The conference itself was a stimulating experience, bringing together a wide variety of people from both industry and academia.

For those not familiar with Hot Chips, a brief introduction might be helpful. Hot Chips tries to offer some windows on the future, from the near future—chips readers can now buy—to futures several years away from market. To maintain this future orientation, the program committee often selects descriptions of chips that are works in progress. Some such chips are clearly research chips not intended for market. Some turn out to be research chips, although they weren't intended to be! Such uncertainty is natural when looking at the future ... so if you attend Hot Chips, please remember that not everything presented actually materializes. (Information on past symposia appears yearly in *IEEE Micro*, beginning with Hot Chips I in the April and June 1990 issues.)

Each year the symposium tries to present an interesting mixture of papers on chips that are really "hot," but in various different directions. Some hot chips really do run at high temperatures, and the program committee looks for several that stretch people's ideas of buildability.

These may well be research prototypes.

Some hot chips are actually "cool"—their interest lies in their ability to pack even more performance and features into smaller and more power-efficient amounts of silicon. This category has become increasingly important.

Some hot chips are hot, not in temperature, but in terms of interest. Either they display instructive research directions, or they represent new generations of widely used chip families.

For this special issue of *IEEE Micro*, I was lucky to be able to obtain articles on the newest, high-end chips from three of the major general-purpose microprocessor families. By now, you may have seen systems based on these chips. The fourth article represents a research chip based on another major architecture. I specifically focused on high-end microprocessors to give you a good opportunity for comparison and contrast of different approaches to similar problems.

The first two articles describe aggressive implementations of existing instruction sets, one CISC, one RISC. Alpert and Avnon describe the Intel Pentium processor. This superscalar processor includes two integer pipelines and one floating-point pipeline, separate instruction and dual-ported data caches, and a branch target buffer. Many of these features are new to this architecture family, and the article analyzes the challenges of matching these features to the required architecture, especially in the area of floating point.

In the second article Asprey et al. describe the Hewlett-Packard PA7100 CPU, the current fastest HP PA-RISC chip. Emphasized here are the various performance features in each of many areas of the design. You might particularly want to study the cache discussion, as HP now seems alone in choosing off-chip primary caches. It is well worth studying the reasoning.

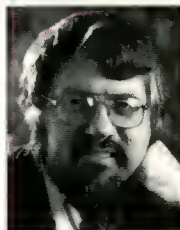
Next McLellan describes a recent RISC architecture, Digital's Alpha AXP, and its first implementation, the 21064. This two-issue implementation emphasized high clock rates, with small on-chip caches and large off-chip secondary caches.

Finally we move forward from leading-edge, commercially available CPUs to a research project. Agarwal et al. describe Sparcle, a Sparc variation designed for research in large-scale multiprocessing. Its goal is not to build the fastest single CPU but to add mechanisms to tolerate longer memory latencies, support fine-grain parallelism, and improve interprocessor communication, all to improve large-scale multiprocessors. Thus, Sparcle illustrates solutions to the challenges of performance improvement, but in a direction orthogonal to that of the previous papers.

This business keeps moving.

HOT CHIPS V WILL TAKE PLACE August 8-10, 1993, at Stanford University in California. The program should be available about the time of this issue's printing. If you'd like more information, contact John Hennessy at (415) 725-3712 or jlh@vsop.stanford.edu.

I thank the authors, who came through with articles on time, and the many other people who worked on this issue, especially Marie English and Dick Price of *IEEE Micro*. ■



John R. Mashey is director, systems technology, at Silicon Graphics, Inc. He works in a wide and rapidly changing range of technical and marketing activities. He has also worked at Bell Laboratories on various Unix-related projects and later contributed to the design of most Mips R-series

RISC chips.

Mashey holds a BS degree in mathematics, and MS and PhD degrees in computer science, all from Pennsylvania State University. He served as an ACM national lecturer for four years and a Usenix program chair, and cofounded the SPEC benchmarking group. He has given more than 400 public talks on Unix, the Programmer's Workbench, software engineering, benchmarking, and the RISC architecture. He is a member of the IEEE Computer Society and the Association of Computing Machinery.

Address questions concerning this special issue to John R. Mashey, Silicon Graphics 7U-005, 2011 North Shoreline Blvd., Mountain View, CA 94039; mash@sgi.com.

Promises, promises ...

... must be kept! And *IEEE Micro* promises to keep you informed on the technical issues that most interest you. Be sure to read

- ✎ **August** for a potpourri of articles on subjects such as cache/user performance
- ✎ **October** for recent activities in the Pacific Rim
- ✎ **December** for standards activities in the IEEE and other standards bodies
- ✎ **Every issue** brings you even more information on microlaw, software/hardware reviews, news, new products, and interviews with industry professionals

Keep current with

IEEE **MICRO**

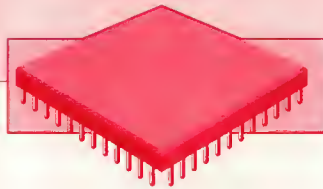
Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 150

Medium 151

High 153



Architecture of the Pentium Microprocessor

The Pentium CPU is the latest in Intel's family of compatible microprocessors. It integrates 3.1 million transistors in 0.8- μ m BiCMOS technology. We describe the techniques of pipelining, superscalar execution, and branch prediction used in the microprocessor's design.

Donald Alpert

Dror Avnon

Intel Corporation

The Pentium processor is Intel's next generation of compatible microprocessors following the popular i486 CPU family. The design started in early 1989 with the primary goal of maximizing performance while preserving software compatibility within the practical constraints of available technology. The Pentium processor integrates 3.1 million transistors in 0.8- μ m BiCMOS technology and carries the Intel trademark. We describe the architecture and development process employed to achieve this goal.

Technology

The continual advancement of semiconductor technology promotes innovation in microprocessor design. Higher levels of integration, made possible by reduced feature sizes and increased interconnection layers, enable designers to deploy additional hardware resources for more parallel computation and deeper pipelining. Faster device speeds lead to higher clock rates and consequently to requirements for larger and more specialized on-chip memory buffers.

Table 1 (next page) summarizes the technology improvements associated with our three most recent microprocessor generations. The 0.8- μ m BiCMOS technology of the Pentium microprocessor enables 2.5 times the number of transistors and twice the clock frequency of the original i486 CPU, which was implemented in 1.0- μ m CMOS.

Compatibility

Since introduction of the 8086 microprocessor in 1978, the X86 architecture has evolved through several generations of substantial functional enhancements and technology improvements, including the 80286 and i386 CPUs. Each of these CPUs was supported by a corresponding floating-point unit. The i486 CPU,¹ introduced in 1989, integrates the complete functionality of an integer processor, floating-point unit, and cache memory into a single circuit.

The X86 architecture greatly appealed to software developers because of its widespread application as the central processor of IBM-compatible personal computers. The success of the architecture in PCs has in turn made the X86 popular for commercial server applications as well. Figure 1 shows some of the well-known software environments that are hosted on the architecture.

The common software environments allow the X86 architecture to exercise several operating modes. Applications developed for DOS use 16-bit real mode (or virtual 8086 mode) and MS Windows. Early versions of OS/2 use 16-bit protected mode, and applications for other popular environments use 32-bit flat (unsegmented) mode. The Pentium microprocessor employs general techniques for improving performance in all operating modes, as well as certain techniques for improving performance in specific operating

Table 1. Technology for microprocessor development.

Microprocessor	Year	Technology	No. of transistors	Frequency (MHz)
i386 CPU	1986	1.5- μ m CMOS, two-layer metal	275K	16
i486 CPU	1989	1.0- μ m CMOS, two-layer metal	1.2M	33
Pentium CPU	1993	0.8- μ m BiCMOS, three-layer metal	3.1M	66

16-bit generation	32-bit generation
DOS	Unix SVR4
MS-Windows	SCO
OS/2 1.x	OSF/1
	Netware 3.11
	Next Step
	32-bit OS/2
	Solaris
	Windows NT
	Univel
	Taligent
1980s	1991
	199x

Figure 1. Software environments. (All figures, tables, and photographs published in this article are the property of Intel Corporation.)

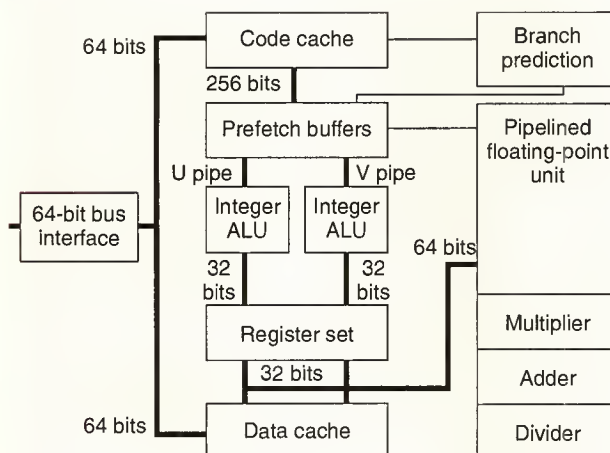


Figure 2. Pentium processor block diagram.

modes. We focus on the 32-bit flat mode here, since this is the most appropriate mode for comparison with the other high-performance microprocessors described at the Hot Chips IV Conference.

The X86 architecture supports the IEEE-754 standard for floating-point arithmetic.² In addition to required operations on single-precision and double-precision formats, the X86 floating-point architecture includes operations on 80-bit, extended-precision format and a set of basic transcendental functions.

Pentium CPU designers found numerous exciting technical challenges in developing a microarchitecture that

maintained compatibility with such a diverse software base. Later in this article we present examples of techniques for supporting self-modifying code and the stack-oriented, floating-point register file.

Performance

A microprocessor's performance is a complex function of many parameters that vary between applications, compilers, and hardware systems. In developing the Pentium microprocessor, the design team addressed these aspects for each of the popular software environments. As a result, Pentium CPU features tuned compilers and cache memory.

We focus on the performance of SPEC benchmarks for both the Pentium microprocessor and i486 CPU in systems with well-tuned compilers and cache memory. More specifically, the Pentium CPU achieves roughly two times the speedup on integer code and up to five times the speedup on floating-point vector code when compared with an i486 CPU of identical clock frequency.

Organization

Figure 2 shows the overall organization of the Pentium microprocessor. The core execution units are two integer pipelines and a floating-point pipeline with dedicated adder, multiplier, and divider. Separate on-chip instruction code and data caches supply the memory demands of the execution units, with a branch target buffer augmenting the instruction cache for dynamic branch prediction. The external interface includes separate address and 64-bit data buses.

Integer pipeline

The Pentium processor's integer pipeline is similar to that of the i486 CPU.³ The pipeline has five stages (see Figure 3) with the following functions:

- *Prefetch.* During the PF stage the CPU prefetches code from the instruction cache and aligns the code to the

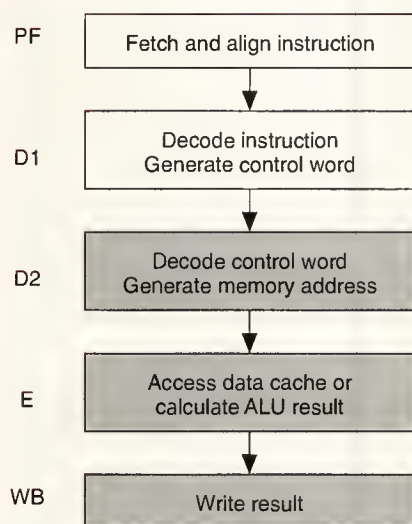


Figure 3. Integer pipeline.

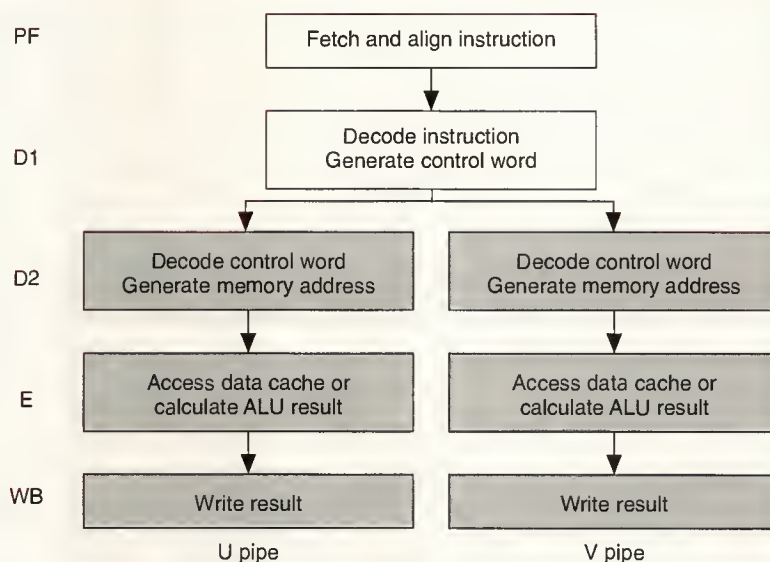


Figure 4. Superscalar execution.

initial byte of the next instruction to be decoded. Because instructions are of variable length, this stage includes buffers to hold both the line containing the instruction being decoded and the next consecutive line.

- *First decode.* In the D1 stage the CPU decodes the instruction to generate a control word. A single control word executes instructions directly; more complex instructions require microcoded control sequencing in D1.
- *Second decode.* In the D2 stage the CPU decodes the control word from D1 for use in the E stage. In addition, the CPU generates addresses for data memory references.
- *Execute.* In the E stage the CPU either accesses the data cache or calculates results in the ALU (arithmetic logic unit), barrel shifter, or other functional units in the data path.
- *Write back.* In the WB stage the CPU updates the registers and flags with the instruction's results. All exceptional conditions must be resolved before an instruction can advance to WB.

Compared to the integer pipeline of the i486 CPU, the Pentium microprocessor integrates additional hardware in several stages to speed instruction execution. For example, the i486 CPU requires two clocks to decode several instruction formats, but the Pentium CPU takes one clock and executes shift and multiply instructions faster. More significantly, the Pentium processor substantially enhances superscalar execution, branch prediction, and cache organization.

Superscalar execution. The Pentium CPU has a superscalar organization that enables two instructions to execute

in parallel. Figure 4 shows that the resources for address generation and ALU functions have been replicated in independent integer pipelines, called U and V. (The pipeline names were selected because U and V were the first two consecutive letters of the alphabet neither of which was the initial of a functional unit in the design partitioning.) In the PF and D1 stages the CPU can fetch and decode two simple instructions in parallel and issue them to the U and V pipelines. Additionally, for complex instructions the CPU in D1 can generate microcode sequences that control both U and V pipelines.

Several techniques are used to resolve dependencies between instructions that might be executed in parallel. Most of the logic is contained in the instruction issue algorithm (see Figure 5) of D1.

```

Decode two consecutive instructions: I1 and I2
If the following are all true
    I1 is a "simple" instruction
    I2 is a "simple" instruction
    I1 is not a jump instruction
    Destination of I1 ≠ source of I2
    Destination of I1 ≠ destination of I2
Then issue I1 to U pipe and I2 to V pipe
Else issue I1 to U pipe
  
```

Figure 5. Instruction issue algorithm.

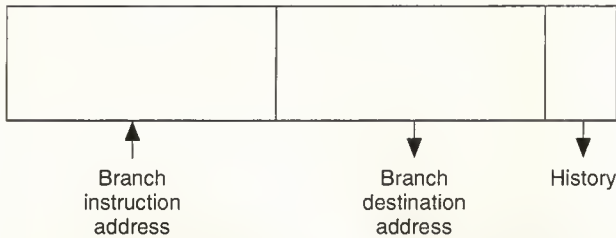


Figure 6. Branch target buffer.

Resource dependencies. A resource dependency occurs when two instructions require a single functional unit or data path. During the D1 stage, the CPU only issues two instructions for parallel execution if both are from a class of "simple" instructions, thereby eliminating most resource dependencies. The instructions must be directly executed, that is, not require microcode sequencing. The instruction being issued to the V pipe can be an ALU operation, memory reference, or jump. The instruction being issued to the U pipe can be from the same categories or from an additional set that uses a functional unit available only in the U pipe, such as the barrel shifter. Although the set of instructions identified as "simple" might seem restrictive, more than 90 percent of instructions executed in the Integer SPEC benchmark suite are simple.

Data dependencies. A data dependency occurs when one instruction writes a result that is read or written by another instruction. Logic in D1 ensures that the source and destination registers of the instruction issued to the V pipe differ from the destination register of the instruction issued to the U pipe. This arrangement eliminates read-after-write (RAW) and write-after-write (WAW) dependencies. Write-after-read (WAR) dependencies need not be checked because reads occur in an earlier stage of the pipelines than writes.

The design includes logic that enables instructions with certain special types of data dependency to be executed in parallel. For example, a conditional branch instruction that tests the flag results can be executed in parallel with a compare instruction that sets the flags.

Control dependencies. A control dependency occurs when the result of one instruction determines whether another instruction will be executed. When a jump instruction is issued to the U pipe, the CPU in D1 never issues an instruction to the V pipe, thereby eliminating control dependencies.

Note that resource dependencies and data dependencies between memory references are not resolved in D1. Dependent memory references can be issued to the two pipelines; we explain their resolution in the description of the data cache.

Branch prediction. The i486 CPU has a simple technique for handling branches. When a branch instruction is executed, the pipeline continues to fetch and decode instructions along the sequential path until the branch reaches the E stage. In E, the CPU fetches the branch destination, and the pipeline resolves whether or not a conditional branch is taken. If the branch is not taken, the CPU discards the fetched destination, and execution proceeds along the sequential path with no delay. If the branch is taken, the fetched destination is used to begin decoding along the target path with two clocks of delay. Taken branches are found to be 15 percent to 20 percent of instructions executed, representing an obvious area for improvement by the Pentium processor.

The Pentium CPU employs a branch target buffer (BTB), which is an associative memory used to improve performance of taken branch instructions (see Figure 6). When a branch instruction is first taken, the CPU allocates an entry in the branch target buffer to associate the branch instruction's address with its destination address and to initialize the history used in the prediction algorithm. As instructions are decoded, the CPU searches the branch target buffer to determine whether it holds an entry for a corresponding branch instruction. When there is a hit, the CPU uses the history to determine whether the branch should be taken. If it should, the microprocessor uses the target address to begin fetching and decoding instructions from the target path. The branch is resolved early in the WB stage, and if the prediction was incorrect, the CPU flushes the pipeline and resumes fetching along the correct path. The CPU updates the dual-ported history in the WB stage. The branch target buffer holds entries for predicting 256 branches in a four-way associative organization.

Using these techniques, the Pentium CPU executes correctly predicted branches with no delay. In addition, conditional branches can be executed in the V pipe paired with a compare or other instruction that sets the flags in the U pipe. Branching executes with full compatibility and no modification to existing software. (We explain aspects of interactions between branch prediction and self-modifying code later.)

Cache organization. The i486 CPU employs a single on-chip cache that is unified for code and data. The single-ported cache is multiplexed on a demand basis between sequential code prefetches of complete lines and data references to individual locations. As just explained, branch targets are prefetched in the E stage, effectively using the same hardware as data memory references. There are potential advantages for such an organization over one that separates code and data.

- 1) For a given size of cache memory, a unified cache has a higher hit rate than separate caches because it balances the total allocation of code and data lines automatically.
- 2) Only one cache needs to be designed.
- 3) Handling self-modifying code can be simpler.

Despite these potential advantages of a unified cache, all of which apply to the i486 CPU, the Pentium microprocessor uses separate code and data caches. The reason is that the superscalar design and branch prediction demand more bandwidth than a unified cache similar to that of the i486 CPU can provide. First, efficient branch prediction requires that the destination of a branch be accessed simultaneously with data references of previous instructions executing in the pipeline. Second, the parallel execution of data memory references requires simultaneous accesses for loads and stores. Third, in the context of the overall Pentium microprocessor design, handling self-modifying code for separate code and data caches is only marginally more complex than for a unified cache.

The instruction cache and data cache are each 8-Kbyte, two-way associative designs with 32-byte lines.

Programs executing on the i486 CPU typically generate more data memory references than when executing on RISC microprocessors. Measurements on Integer SPEC benchmarks show 0.5 to 0.6 data references per instruction for the i486 CPU⁴ and only 0.17 to 0.33 for the Mips processor.⁵ This difference results directly from the limited number (eight) of registers for the X86 architecture, as well as procedure-calling conventions that require passing all parameters in memory. A small data cache is adequate to capture the locality of the additional references. (After all, the additional references have sufficient locality to fit in the register file of the RISC microprocessors.) The Pentium microprocessor implements a data cache that supports dual accesses by the U pipe and V pipe to provide additional bandwidth and simplify compiler instruction scheduling algorithms.

Figure 7 shows that the address path to the translation look-aside buffer and data cache tags is a fully dual-ported structure. The data path, however, is single ported with eight-way interleaving of 32-bit-wide banks. When a bank conflict occurs, the U pipe assumes priority, and the V pipe stalls for a clock cycle. The bank conflict logic also serves to eliminate data dependencies between parallel memory references to a single location. For memory references to double-precision floating-point data, the CPU accesses consecutive banks in parallel, forming a single 64-bit path.

The design team considered a fully dual-ported structure for the data cache, but feasibility studies and performance simulations showed the interleaved structure to be more effective. The dual-ported structure eliminated bank conflicts, but the SRAM cell would have been larger than the cell used in the interleaved scheme, resulting in a smaller cache and lower hit ratio for the allocated area. Additionally, the handling of data dependencies would have been more complex.

With a write-through cache-consistency protocol and 32-bit data bus, the i486DX2 CPU uses buses 80 percent of the time; 85 percent of all bus cycles are writes. (The i486DX2 CPU has a core pipeline that operates at twice the bus clock's

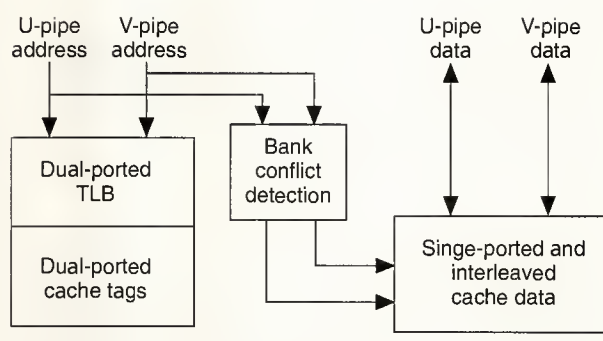


Figure 7. Dual-access data cache.

frequency.) For the Pentium microprocessor, with its higher performance core pipelines and 64-bit data bus, using a write-back protocol for cache consistency was an obvious enhancement. The write-back protocol uses four states: modified, exclusive, shared, and invalid (MESI).

Self-modifying code. One challenging aspect of the Pentium microprocessor's design was supporting self-modifying code compatibly. Compatibility requires that when an instruction is modified followed by execution of a taken branch instruction, subsequent executions of the modified instruction must use the updated value. This is a special form of dependency between data stores and instruction fetches.

The interaction between branch predictions and self-modifying code requires the most attention. The Pentium CPU fetches the target of a taken branch before previous instructions have completed stores, so dedicated logic checks for such conditions in the pipeline and flushes incorrectly fetched instructions when necessary. The CPU thoroughly verifies predicted branches to handle cases in which an instruction entered in the branch target buffer might be modified. The same mechanisms used for consistency with external memory maintain consistency between the code cache and data cache.

Floating-point pipeline

The i486 CPU integrated the floating-point unit (FPU) on chip, thus eliminating overhead of the communication protocol that resulted from using a coprocessor. Bringing the FPU on chip substantially boosted performance in the i486 CPU. Nevertheless, due to limited devices available for the FPU, its microarchitecture was based on a partial multiplier array and a shift-and-add data path controlled by microcode. Floating-point operations could not be pipelined with any other floating-point operations; that is, once a floating-point instruction is invoked, all other floating-point instructions stall until its completion.

The larger transistor budget available for the Pentium microprocessor permits a completely new approach in the design of the floating-point microarchitecture. The aggressive

Integer pipe	PF	D1	D2	E	WB			
Floating-point pipe	PF	D1	D2	E	X1	X2	WF	ER

Figure 8. Floating-point pipeline.

performance goals for the FPU presented an exciting challenge for the designers, even with more silicon resources available. Furthermore, maintaining full compatibility with previous products and with the IEEE standard for floating-point arithmetic was an uncompromising requirement.

Floating-point pipeline stages. Pentium's floating-point pipeline consists of eight stages. The first two stages are processed by the common (integer pipeline) resources for prefetch and decode. In the third stage the floating-point hardware begins activating logic for instruction execution. All of the first five stages are matched with their counterpart integer pipeline stages for pipeline sequencing and synchronization (see Figure 8).

- *Prefetch.* The PF stage is the same as in the integer pipeline.
- *First decode.* The D1 stage is the same as in the integer pipeline.
- *Second decode.* The D2 stage is the same as in the integer pipeline.
- *Operand fetch.* In this E stage the FPU accesses both the data cache and the floating-point register file to fetch the operands necessary for the operation. When floating-point data is to be written to the data cache, the FPU converts internal data format into the appropriate memory representation. This stage matches the E stage of the integer pipeline.
- *First execute.* In the X1 stage the FPU executes the first steps of the floating-point computation. When floating-point data is read from the data cache, the FPU writes the incoming data into the floating-point register file.
- *Second execute.* In the X2 stage the FPU continues to execute the floating-point computation.
- *Write float.* In the WF stage the FPU completes the execution of the floating-point computation and writes the result into the floating-point register file.
- *Error reporting.* In the ER stage the FPU reports internal special situations that might require additional processing to complete execution and updates the floating-point status word.

The eight-stage pipeline in the FPU allows a single cycle throughput for most of the "basic" floating-point instructions such as floating-point add, subtract, multiply, and compare. This means that a sequence of basic floating-point instructions free from data dependencies would execute at a rate of

one instruction per cycle, assuming instruction cache and data cache hits.

Data dependencies exist between floating-point instructions when a subsequent instruction uses the result of a preceding instruction. Since the actual computation of floating-point results takes place during X1, X2, and WF stages, special paths in the hardware allow other stages to be bypassed and present the result to the subsequent instruction upon generation. Consequently, the latency of the basic floating-point instructions is three cycles.

The X86 floating-point architecture supports single-precision (32-bit), double-precision (64-bit), and extended-precision (80-bit) floating-point operations. We chose to support all computation for the three precisions directly, by extending the data path width to support extended precision. Although this entailed using more devices for the implementation, it greatly simplified the microarchitecture while improving the performance. If smaller data paths were designed, special rerouting of the data within the FPU and several state machines or microcode sequencing would have been required for calculating the higher precision data.

Floating-point instructions execute in the U pipe and generally cannot be paired with any other integer or floating-point instructions (the one exception will be explained later). The design was tuned for instructions that use one 64-bit operand in memory with the other operand residing in the floating-point register file. Thus, these operations may execute at the maximum throughput rate, since a full stage (E stage) in the pipeline is dedicated to operand fetching. Although floating-point instructions use the U pipe during the E stage, the two ports to the data cache (which are used by the U pipe and the V pipe for integer operations) are used to bring 64-bit data to the FPU. Consequently, during intensive floating-point computation programs, the data cache access ports of the U pipe and V pipe operate concurrently with the floating-point computation. This behavior is similar to superscalar load-store RISC designs where load instructions execute in parallel with floating-point operations, and therefore deliver equivalent throughput of floating-point operations per cycle.

Microarchitecture overview. The floating-point unit of the Pentium microprocessor consists of six functional sections (see Figure 9).

The floating-point interface, register file, and control (FIRC) section is the only interface between the FPU and the rest of the CPU. Since the function of floating-point operations is usually self-contained within the floating-point computation core, concentrating all the interface logic in one section helped to create a modular design of the other sections. The FIRC section also contains most of the common floating-point resources: register file, centralized control logic, and safe instruction recognition logic (described later). FIRC can complete execution of instructions that do not need arithmetic compu-

tation. It dispatches the instructions requiring arithmetic computation to the arithmetic sections.

The floating-point exponent section (FEXP) calculates the exponent and the sign results for all the floating-point arithmetic operations. It interfaces with all the other arithmetic sections for all the necessary adjustments between the mantissa and the sign-and-exponent fields in the computation of floating-point results.

The floating-point multiplier section (FMUL) includes a full multiplier array to support single-precision (24-bit mantissa), double-precision (53-bit mantissa), and extended-precision (64-bit mantissa) multiplication and rounding within three cycles. FMUL executes all the floating-point multiplication operations. It is also used for integer multiplication, which is implemented through microcode control.

The floating-point adder section (FADD) executes all the "add" floating-point instructions, such as floating-point add, subtract, and compare. FADD also executes a large set of micro-operations that are used by microcode sequences in the calculation of complex instructions, such as binary coded decimal (BCD) operations, format conversions, and transcendental functions. The FADD section operates during the X1 and X2 stages of the floating-point pipeline and employs several wide adders and shifters to support high-speed arithmetic algorithms while maintaining maximum performance for all data precisions. The CPU achieves a latency of three cycles with a throughput of one cycle for all the operations directly executed by the FADD section for single-precision, double-precision, and extended-precision data.

The floating-point divider (FDIV) section executes the floating-point divide, remainder, and square-root instructions. It operates during the X1 and X2 pipeline stages and calculates two bits of the divide quotient every cycle. The overall instruction latency depends on the precision of the operation. FDIV uses its own sequencer for iterative computation during the X1 stage. The results are fully accurate in accordance with IEEE standard 754 and ready for rounding at the end of the X2 stage.

The floating-point rounder (FRND) section rounds the results delivered from the FADD and FDIV sections. It operates during the WF stage of the floating-point pipeline and delivers a rounded result according to the precision control and the rounding control, which are specified in the floating-point control word.

Safe instruction recognition. Floating-point computation requires longer execution times than integer computation. Pentium's floating-point pipeline uses eight stages, while the integer pipeline uses only five stages. Compatibility requires in-order instruction execution as well as precise exception reporting. To meet these requirements in the Pentium processor, floating-point instructions should not proceed beyond the X1 stage, that is, allow subsequent instructions to proceed beyond the E stage, unless the floating-point instruction is guaranteed to complete without causing an ex-

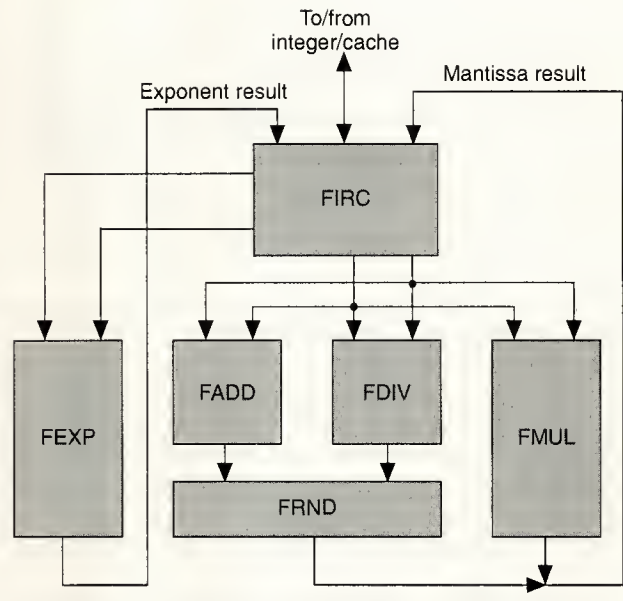


Figure 9. Floating-point unit block diagram.

ception. Otherwise, an instruction may change the state of the CPU, while an earlier floating-point instruction (which has not yet completed) might cause an exception that requires a trap to a software exception handler.

To avoid a substantial performance loss due to stalling instructions until the exception status of a previous floating-point instruction is known, Pentium's floating-point unit employs a mechanism called safe instruction recognition (SIR). This logic determines whether a floating-point instruction is guaranteed to complete without creating an exception and therefore is considered "safe." If an instruction is safe, there is no need to stall the pipeline, and the maximum throughput can be obtained. If, however, the instruction is not safe, the pipeline stalls for three cycles until the unsafe instruction reaches the ER stage and a final determination of the exception status is made.

Six possible exceptions can occur on the Pentium microprocessor's floating-point operations: invalid operation, divide by zero, denormal operand, overflow, underflow, and inexact. The SIR logic needs to determine early in the floating pipeline—in the X1 stage—before any computation takes place whether the instruction is guaranteed to be exception free (safe) or not (unsafe). The first three of the six exceptions can be detected without any floating-point calculation. From the latter three exceptions, the inexact exception is usually "masked" by the operating system or the software application (using the precision mask, or PM, bit in the floating-point control word). Otherwise, a trap will occur whenever rounding of the result is necessary. When the pre-

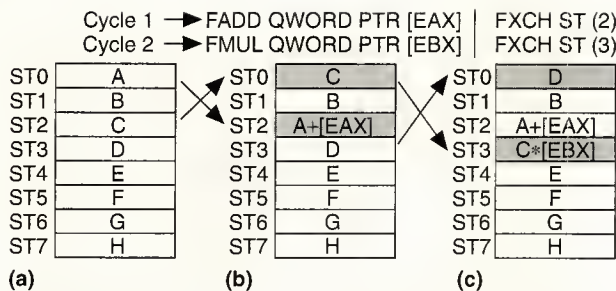


Figure 10. FXCH code example.

cision (inexact) exception is masked, the pipeline delivers the correctly rounded result directly. For overflow and underflow exceptions SIR logic uses an algorithm that monitors the exponent fields of the input operands to conclude the exception status (safe or unsafe).

In the X86 architecture the CPU stores floating-point operands in the floating-point register file with an extended-precision exponent, regardless of the precision control in the floating-point control word. The extended-precision exponent supports much greater range than the double-precision format. Overflow and underflow exceptions caused by converting the data into double-precision or single-precision formats occur only when storing the data into external memory. These characteristics of the X86 floating-point architecture give a unique advantage to the effectiveness of the SIR mechanism in the Pentium CPU, since the SIR algorithm can use the internal (extended-precision) exponent range. Thus, the occurrence of unsafe operations is extremely rare. Our evaluation of the SIR algorithm for the FPU design found no unsafe instructions in simulated execution of the SPEC89 floating-point benchmarks.

Register stack manipulation. The X86 floating-point instruction set uses the register file as a stack of eight registers in which the top of stack (TOS) acts as an accumulator of the results. Therefore, the top of the stack is used for the majority of the instructions as one of the source operands and, usually, as the destination register.

To improve the floating-point pipeline performance by optimizing the use of the floating-point register file, Pentium's FPU can execute the FXCH instruction in parallel with any basic floating-point operation. The FXCH instruction "swaps" the contents of the TOS register with another register in the floating-point register file. All the basic floating-point instructions may be paired with FXCH in the V pipe. The pair execute in parallel, even when data dependency between the two instructions in the pair exists. The use of parallel FXCH redirects the result of a floating-point operation to any selected register in the register file, while bringing a new operand to the top of the stack for immediate use by the next floating-point operation.

The example shown in Figure 10 illustrates the use of parallel FXCH. The code in the example generates the results of two independent floating-point calculations. The floating-point register file contains initial values prior to code execution: register ST0 (TOS) contains the value A, register ST1 contains value B, register ST2 contains value C, and so on. The two operations are

- 1) floating-point addition of value A with the 64-bit floating-point operand addressed by the general register EAX, and
- 2) floating-point multiplication of value C by the 64-bit floating-point operand addressed by the general register EBX.

When the floating-point pipeline is fully loaded and these two operations are part of the code sequence, the parallel FXCH allows the calculation to maintain the maximum throughput of one cycle per operation. Within one cycle the Pentium CPU writes the result of the addition to ST2, while the operand for the next operation moves to the top of the stack. On the next cycle, the processor writes the result of the multiplication to ST3, while the top of the stack contains value D, which may be used for a subsequent operation.

Transcendental instructions. The CPU supports all eight transcendental instructions that are defined in the instruction set through direct execution of microcode sequences. The transcendental instructions are

- | | |
|------------|-----------------------|
| 1) FSIN | sine, |
| 2) FCOS | cosine, |
| 3) FSINCOS | sine and cosine, |
| 4) FPTAN | tangent, |
| 5) FPATAN | arctangent, |
| 6) F2XM1 | $2^X - 1$, |
| 7) FYL2X | $Y * \log_2(X)$, and |
| 8) FYL2XP | $1 Y * \log_2(X+1)$ |

We developed new, table-driven algorithms for the transcendental functions using polynomial approximation techniques. These algorithms substantially improved performance and accuracy over the i486 CPU implementation, which used the more traditional Cordic algorithms. The approximation tables reside in an on-chip ROM along with the other special constants that are used for floating-point computation.

The performance improvement of the transcendental instructions on the Pentium processor ranges from two to three times over the same instructions on the i486 CPU at the same frequency. The worst-case error for all the transcendental instructions is less than 1 ulp (unit in the last place) when rounding to nearest even and less than 1.5 ulps when rounding in other modes. The functions are guaranteed to be monotonic, with respect to the input operands, throughout the domain supported by the instruction.

Development process

Developing a highly integrated microprocessor involves collaboration between numerous teams having diverse technical specialties and working under the discipline of well-defined methodologies. A small team of architects and VLSI designers developed the initial concepts of the design. This group conducted feasibility studies of parallel instruction decoding and options for branch prediction techniques. Simultaneously, it evaluated performance by hand for short benchmarks and compiler optimizations. As initial directions were established, additional engineers participated, and subteams focused on the following areas:

- 1) behavioral modeling of the microarchitecture;
- 2) circuit feasibility design for caches, decoding PLAs (programmable logic arrays), floating-point data path, and other critical functions;
- 3) a flexible, trace-driven simulator of instruction timing for performance evaluation;
- 4) a prototype compiler; and
- 5) enhancements to existing instruction-tracing tools.

Throughout the design we refined the Pentium microprocessor using both top-down and bottom-up methods. Top-down refinement was accomplished through comprehensive characterization of executing benchmark work loads on the i486 CPU⁴ and trace-driven experiments concerning alternative machine organizations conducted by architects using the performance simulator.

VLSI design engineers evaluating features critical to the targeted area and frequency refined the design from the bottom up. On two occasions in the design the accumulation of changes from bottom-up refinement caused the need for substantial restructuring of the microprocessor's global chip plan, or "die diets." On those occasions, interdisciplinary teams of specialists collaborated to brainstorm and evaluate ideas that could satisfy the global or local design constraints. In one instance, we found it necessary to refine the set of instructions that could be executed in parallel. Constraints had been assigned to the area and speed of the decoder PLAs. The VLSI designers identified combinations of instruction formats that would feasibly decode in parallel, and the compiler writers determined the optimal selection.

In the end, the measured performance of the Pentium microprocessor in production systems is within 2 percent of that predicted before the design was completed.

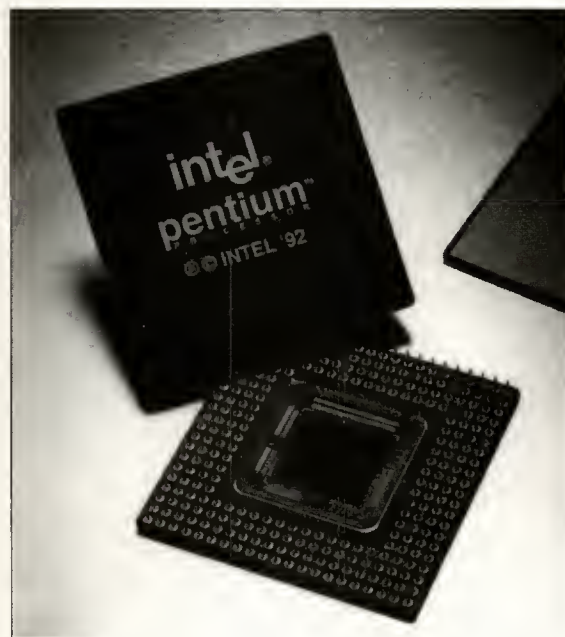
The logic validation of the Pentium processor design presented a major challenge to the design team. A comprehensive test base from the validation of previous X86 microprocessors was available. However, the Pentium processor microarchitecture introduced several new fundamental techniques, such as superscalar, write-back cache, and floating-point algorithms, that required a more rigorous veri-

Naming the Pentium processor

In naming the fifth generation of its compatible microprocessor line the Pentium processor, Intel departed from tradition. Pentium breaks a string of CPU products dating back to the late 1970s that used numerics (8086, 286, 386, 486).

"The natural course would be to call this chip the 586," said Andrew S. Grove, president and chief executive officer. "Unfortunately, we cannot trademark those numbers, which means that any company might call any chip a 586, even if it doesn't measure up to the real thing."

Pentium uses the Greek word for five, "pente," as its root to associate with the fifth-generation product and adds "-ium," a common ending from the periodic table of elements. Thus, the Pentium microprocessor is the fifth generation, a key element for future computing.



fication methodology.

We used different validation approaches in pre-silicon testing of the Pentium microprocessor:

- 1) Architecture verification looked at the "black box" functionality from the programmer's point of view. We designed comprehensive tests to cover all possible aspects of the programming model and all the Pentium processor user-visible features.

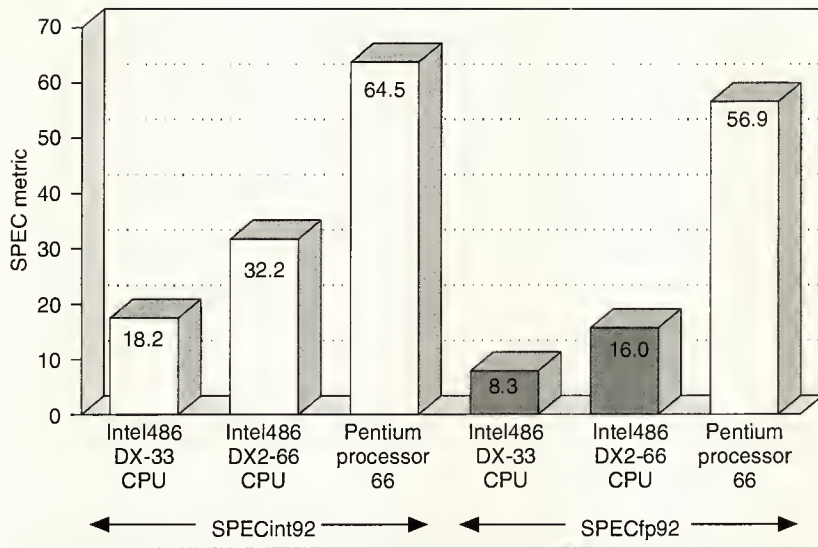


Figure 11. Pentium processor and i486 CPU performance for SPEC benchmarks.

- 2) Design verification checked the internal functionality from the point of view of a logic designer who would understand the behavior of every internal signal. This testing approach is considered a "white box" technique, in which tests are written to exercise all the internal logic and verify its correct behavior.
- 3) Random instruction testing was a valuable tool to cover all those situations that are rarely covered by the more traditional, handwritten tests. Running finely tuned random tests let us verify correct functionality by comparing the results generated by a logic design description of the Pentium processor to the results generated by a software-emulated model.
- 4) A logic-design hardware model (QuickTurn) enabled increased testing coverage capacity by allowing a much larger software base to run on the processor model before the first silicon was available. We ported the logic model of the Pentium processor onto a QuickTurn setup, which was capable of handling the complete design, and tested major operating systems and application programs before finalizing the design.

In addition to the general validation approach, we dedicated a special effort to verify the new algorithms employed by the FPU. We developed a high-level software simulator to evaluate the intricacies of the specific add, multiply, and divide algorithms used in the design. This simulator then evolved into a testing environment, allowing the verification of the FPU logic design model independently from the rest of the Pentium processor. Also, the new algorithms used for the

floating-point transcendental functions required an extensive test strategy that verified the accuracy and monotonicity of the results throughout the development process, comparing the results to a "super accurate" software model. Eventually, when the first silicon of the Pentium processor was available for testing, we used automatic testing techniques to assure the correctness of the transcendental instructions.

Compiler optimizations

The compiler technology developed with the Pentium microprocessor includes machine-independent optimizations common to current high-performance compilers, such as inlining, unrolling, and other loop transformations. In addition, we used techniques specifically developed for the X86 architecture and tuned them for the Pentium processor's microarchitecture.

The X86 architecture has certain characteristics that require specialized optimization techniques different from those for RISC architectures. The architecture supports a variety of instruction formats for equivalent operations. Consequently, it is critical to select instruction formats that are decoded most efficiently by the processor. The X86 register set includes only eight integer and eight floating-point registers. We have found that common global register allocation techniques that assign variables to registers for the entire scope of a procedure are ineffective with such a limited number of registers. Registers must be allocated within a narrower scope and together with instruction scheduling.

The compiler schedules instructions to minimize interlocks and to maximize parallel execution for the Pentium processor's superscalar pipelines. These techniques also benefit performance on the i486 CPU (though to a lesser extent) because the processors' pipeline organizations are similar. The instruction-scheduling techniques have minimal impact on performance for the i386 CPU since that processor uses little pipelining. As explained in the description of the floating-point pipeline, the compiler schedules FXCH instructions to avoid floating-point register-stack dependencies.

THE PENTIUM MICROPROCESSOR employs superscalar integer pipelines, branch prediction, and a highly pipelined FPU to achieve the highest X86 performance levels available elsewhere while preserving binary compatibility with the X86 architecture. Figure 11 summarizes the performance of the Pentium microprocessor and the highest performance i486

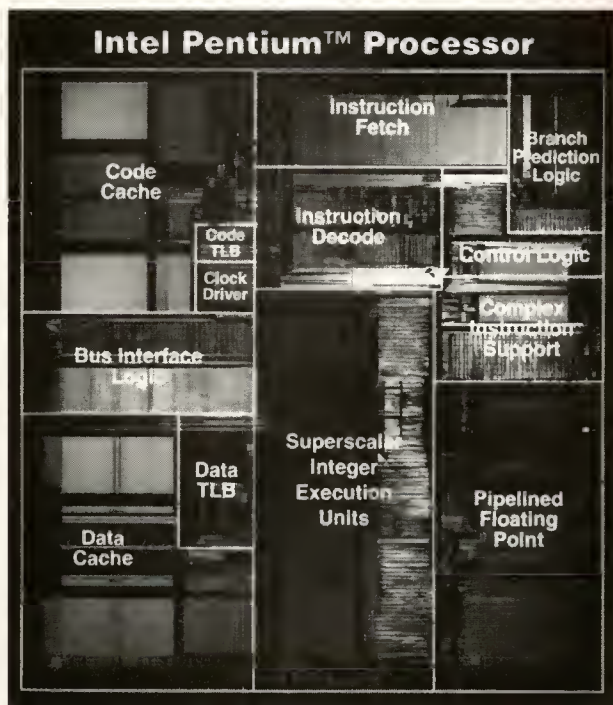


Figure 12. Die photograph.

CPU for the SPEC benchmarks in well-tuned systems. Figure 12 reproduces a photograph of the packaged circuit that integrates 3.1 million transistors. ■

Acknowledgments

The individuals who made substantial contributions to the Pentium processor's design are too numerous to list here, so instead we acknowledge groups of contributors. The VLSI design team applied their creativity and determined effort throughout the project. The compiler team developed and implemented novel optimization techniques. Software engineers in several groups developed instruction-tracing and performance simulation tools. Hardware engineers and technicians instrumented measurement and tracing systems. Architects facilitated and integrated efforts of these other teams. The efforts in architecture, optimizing compiler, and performance simulation involved collaboration between teams in Santa Clara and Israel.

References

1. *i486 Processor Programmer's Reference Manual*, Intel Corporation, Santa Clara, Calif., 1990.
2. *ANSI/IEEE Standard 754-1985 for Binary Floating-Point Arithmetic*, IEEE Computer Society Press, Los Alamitos, Calif., 1985.

3. John H. Crawford, "The i486 CPU: Executing Instructions in One Clock Cycle," *IEEE Micro*, Vol. 10, No. 1, Feb., 1990, pp. 27-36.
4. Tejpal Chadha and Partha Srinivasan, "The Intel386 CPU Family—Architecture & Performance Analysis," *Digest of Papers Comcon Spring 1992*, CS Press, Feb. 1992, pp. 332-337.
5. Robert F. Cmelik et al., "An Analysis of Mips and Sparc Instruction Set Utilization on the SPEC Benchmarks," *Proc. ASPLOS-IV Conf., Computer Architecture News*, Vol. 19, No. 2, Apr., 1991, pp. 290-302.



Donald Alpert is an architecture manager in Intel Corporation's Microprocessor Division. He holds responsibility for managing the architecture team that developed specifications and modeling and evaluating performance of the Pentium processor. Previously, he held various microprocessor development positions at National Semiconductor Corporation and Zilog.

Alpert received a BS degree from MIT and MS and PhD degrees from Stanford University, all in electrical engineering. He is a member of the IEEE Computer Society and the Association of Computing Machinery.



Dror Avnon is design manager of the floating-point unit of the Pentium processor. He holds responsibility for the micro-architecture, design, performance analysis, and verification for the FPU logic and microcode. He previously held design engineering positions at National Semiconductor Corporation, Computer Consoles, and Elscint.

Avnon received a BSc degree in electronic engineering from Technion-Israel Institute of Technology in Haifa. He is a member of the IEEE Computer Society.

Direct questions to Dror Avnon, Intel Corporation, M/S RN2-27, 2200 Mission College Blvd., Santa Clara, CA 95052; davnon@mipos2.intel.com.

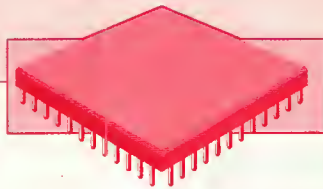
Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 159

Medium 160

High 161



Performance Features of the PA7100 Microprocessor

The PA7100 CPU is the first PA-RISC implementation to combine an integer core and floating-point coprocessor into a single-chip format. It also incorporates superscalar execution and supports clock rates of up to 100 MHz in standard 0.8- μ m CMOS. Features such as a flexible primary cache organization and multiprocessing capability allow the device to scale into a variety of system applications, price ranges, and performance levels.

Tom Asprey

Gregory S. Averill

Eric DeLano

Russ Mason

Bill Weiner

Jeff Yetter

Hewlett-Packard

The PA7100 CPU is the seventh implementation of the Hewlett-Packard precision architecture, reduced instruction-set computer (PA-RISC) architecture. Since the first transistor-transistor logic PA-RISC was introduced in 1986, performance has doubled roughly every 18 months in subsequent implementations. As shown in Figure 1, the PA7100 has continued this performance growth trend with its introduction in 1992.

Time-to-market and binary compatibility with existing PA-RISC processors were important considerations in the design of the PA7100. Thus our design adapted much of the integer core, memory management circuitry, and cache interfaces from the earlier 66-MHz PA-RISC CPU.^{1,2} See Figure 2 for a photograph of the PA7100 die. To extend the clock to a frequency of 100 MHz, we employed a combination of design evolution and improvements in CMOS IC technology. Measuring 1.42×1.42 cm, this chip uses the 0.8- μ m CMOS26B process.

The performance goals of the PA7100 required that new features and circuits be developed in several key areas.^{3,4} Most notable is a new floating-point coprocessor that is included on the chip and that delivers exceptional performance. Occupying less than 30 square millimeters of silicon area, the PA7100 coprocessor achieves an exceptionally low arithmetic latency. A new superscalar execution model allows the simultaneous

dispatch of integer and floating-point instructions to further improve coprocessor performance. We improved the cache and translation look-aside buffer (TLB) subsystems, and also the interface to main memory. The PA7100 CPU retained hardware support for shared-memory multiprocessing. Our design adds a new two-way multiprocessing interconnection protocol that allows the CPU to scale into previously unavailable low-cost microprocessor applications.

Like all of the preceding PA-RISC implementations, the PA7100 relies on external static RAM arrays for primary cache. This arrangement allows system designers to configure data caches up to 2 Mbytes and instruction caches up to 1 Mbyte as well as scaled-down configurations in a wide range of price and performance levels.

While the primary design focus for the PA7100 was the high-performance technical desktop, the single-chip format lets us extend the architecture into the lowest price points yet available. The chip's unique set of features suits it equally well for a range of commercial applications. For example, a new type of hierarchical TLB known as a hardware table walker provides the best performance in virtual memory management of any PA-RISC to date.

To fully describe the PA7100 CPU, we first must discuss each of its major subsystems, emphasizing especially the design features and capabilities that support performance or scalability.

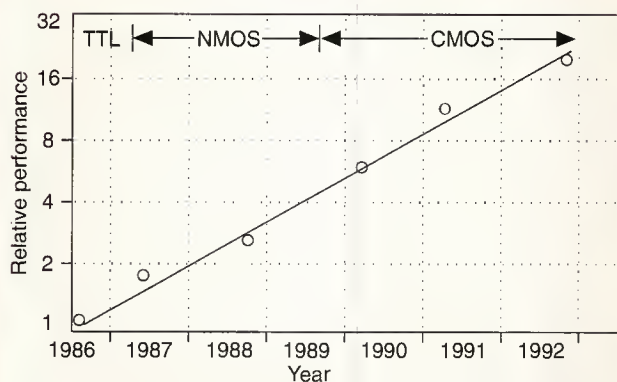


Figure 1. Quest for performance improvement.

Instruction execution

The instruction pipeline reflects many fundamental design choices. The most important of these choices is the cache structure. Although the use of on-chip rather than off-chip caches can facilitate faster pipeline clock rates, on-chip caches are usually not large enough to achieve acceptably low cache miss rates. For this reason, and also to provide for a high degree of price-to-performance scalability, the PA7100 employs off-chip caches made from industry-standard SRAMs.

By dedicating three of the five pipeline cycles solely to cache access, we designed the PA7100 pipeline and system components so that only the cache SRAM read access time would limit the pipeline clock rate. The off-chip caches are cycled at the pipeline frequency of 100 MHz, and the chip can execute a load instruction every cycle. To maximize the processor frequency, the design relaxes write timing to two cycles so that this timing does not become critical. Single-cycle write operations would have reduced the processor frequency because of the time required to tri-state the SRAM outputs between reads and writes.

Figure 3 shows how our design executes various types of instructions in the pipeline. Each stage of the pipeline is divided into two equal phases, with the first three phases of the pipeline dedicated to instruction fetching. The next two phases include decode and data cache address generation. Next come three phases of data cache access (for loads and stores), and the last stage includes register write-back. The execution of integer operations, floating-point opera-

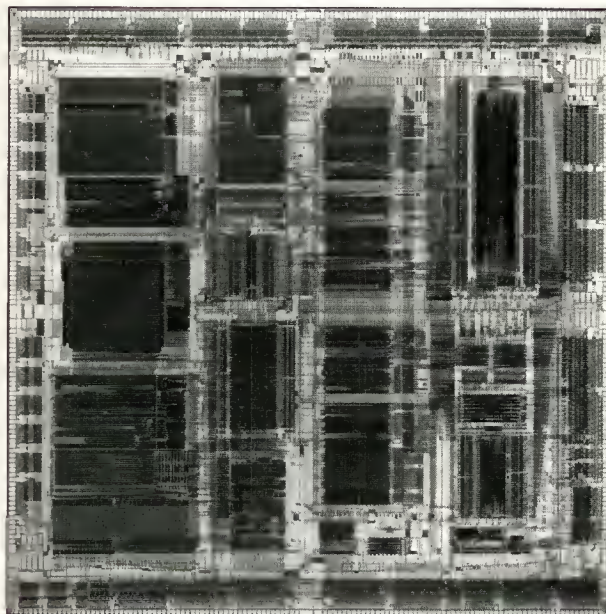


Figure 2. The PA7100.

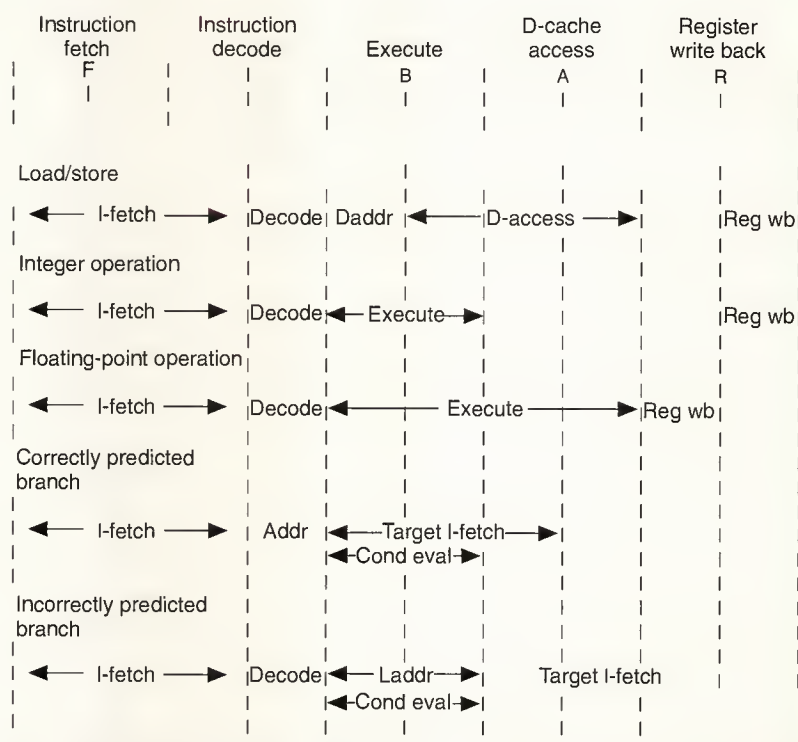


Figure 3. The instruction execution pipeline.

tions, and branches is straightforward. These operations are also shown in the figure.

Instruction execution pipeline

The partitioning of instruction execution into pipeline stages involves trade-offs between the clock rate and the number of pipeline interlock/branch penalties that are incurred by code. To increase the frequency of a pipeline, execution must be spread out over more stages, which increases the number of pipeline interlocks due to data/fetch dependencies. This helps explain why a 200-MHz computer does not always perform better than a 100-MHz computer. The PA7100 minimizes all pipeline interlock penalties subject to the constraint that the read cycle time of the cache RAMs determines the pipeline frequency. Table 1 shows some of the pipeline interlocks for the PA7100, the DEC Alpha processor, the Mips R4000, and the IBM RS/6000.^{5,6} As expected, the PA7100 has fewer interlocks than the corresponding DEC Alpha. The effect of pipeline interlocks on overall performance depends on the benchmark, the compiler, and the memory system, issues that are beyond the scope of this discussion.

The PA7100 floating-point unit produces results with very low latencies. Most notably, the floating-point add/subtract,

and multiply units have only two-cycle latency, and add/subtract/multiply instructions can be issued every cycle. A very high floating-point performance results, as does a simplified pipeline control design. In addition, the processor supports the superscalar execution of all floating-point operations with integer operations or integer and floating-point loads and stores. There are also no order or alignment constraints on the pair of instructions that are executed together.

The PA7100 implements a simple static branch prediction algorithm to allow for a zero-cycle branch penalty. This algorithm predicts that forward conditional branches are untaken and that backward branches are taken. Other current microprocessor designs have resorted to branch target caches and speculative execution to minimize their branch penalties, measures that are warranted only when the maximum branch penalty is high.⁷ The PA-RISC architecture also provides inherent parallelism for conditional branches.⁸ The conditional branches in the PA-RISC architecture perform operations such as compare, add, and move in parallel with the branch target calculation, and the branch condition is based on this operation.

The PA7100 implements a one-entry store buffer to minimize the store penalty. The chip writes the store buffer to cache at the same time that it performs the read tag operation for the next store instruction. This implementation has a maximum store penalty of one cycle. We can avoid this penalty simply by scheduling a non-load/store instruction immediately after the store instruction. Also note that because integer and floating-point results are calculated so quickly almost no benefit would come from implementing out-of-order stores or branches that wait for previous results.

Executing instructions in the PA7100 then is relatively simple and straightforward. The chip has to contend with very few pipeline interlocks. Nor does it require complex techniques to achieve a small number of average cycles per instruction. Because the pipeline interlocks in the PA7100 are so rare, the cycles lost to cache misses and TLB misses constitute the largest portion of the average number of cycles per instruction beyond the baseline cycles due to the pipeline. To minimize these cycles, the PA7100 has implemented an assortment of techniques and features that we will soon describe.

Caches

A major contribution to the performance of the PA7100 came from increasing the processor frequency to 100 MHz. Although we leveraged much of the cache design directly from the previous 66-MHz PA-RISC implementation used in our original Series 700 products, the PA7100 cache design required extensive changes. Reaching the required performance levels demanded more than an improved CMOS process and faster SRAMs.

Because the PA7100's cache load access time determines the maximum pipeline speed, the cache needed a careful

Table 1. Pipeline interlocks for various microprocessors (penalties in cycles rather than instructions).

	PA7100	DEC Alpha	IBM RS/6000	Mips R4000
Maximum branch penalty	1	4	3	2
Maximum load use penalty	1	2/6*	1	2**
Maximum integer ALU interlock	0	1	0	0
Maximum floating-point ALU interlock	1	5	1	3
Maximum floating-point multiply interlock	1	5	1	6/7†
Maximum pipeline frequency (MHz)	100	200	62.5	100

* 2 for on-chip hit, 6 for off-chip hit
 ** 2 for on-chip hit, external cache operates at 25 MHz
 † Single precision/double precision

design approach. Not only did we need to allow for a high processor frequency but we also had to supply the number of data words and instructions per cycle that a very high performance RISC processor like the PA7100 requires. Reading two instructions per cycle allowed for superscalar execution, while writing two instructions at a time reduced penalties for I-cache misses. Reading and writing two data words at a time made double-word loads and stores possible, maximizing transfer rates between the CPU's registers and the data cache.

In addition to the speed and performance concerns already discussed, cost and scalability were also major considerations. The PA7100 was intended for use in systems that ranged from low-cost desktop computers to high-performance compute servers, super minicomputers, and large parallel arrays for supercomputers. Each of these applications has its own requirements for performance and cost. Low-end systems usually emphasize cost by reducing the size and speed of their caches. For high-end system designs, where performance is the dominating factor, a premium for larger and faster caches with higher performance is usually acceptable.

The PA7100 design addressed these issues by implementing its cache memory with separate external instruction and data caches connected to the CPU in a Harvard configuration. As shown in Figure 4, each cache connects to the CPU via its own independent 64-bit data path. The CPU then can read two data words and two instructions every cycle or write them every two cycles. At 100 MHz, this gives each bus a read bandwidth of 800 Mbytes/s and a write bandwidth of 400 Mbytes/s, giving the PA7100 cache a higher level of performance than most other RISC processors' primary "on-chip" caches.

The instruction and data caches are both virtually addressed. Data and instructions transfer to and from memory in eight-word cache lines. Both caches are directly mapped and scalable. The size of the instruction cache can be configured from 4 Kbytes to 1 Mbyte, while the data cache has a range of 4 Kbytes to 2 Mbytes. Using SRAMs widely available today, these primary external caches are vastly larger than the primary caches that can be reasonably fabricated on a single-chip CPU using current technologies. Table 2 lists the various cache configuration options and SRAM requirements. External caches also do not waste precious CPU real estate. Also, because our design uses industry-standard asynchronous SRAMs, memory can be upgraded at the lowest cost possible with newer memory technology as it becomes available. Keeping the primary caches off chip also allows for a range of configurations to provide for products at varying levels of price and performance.

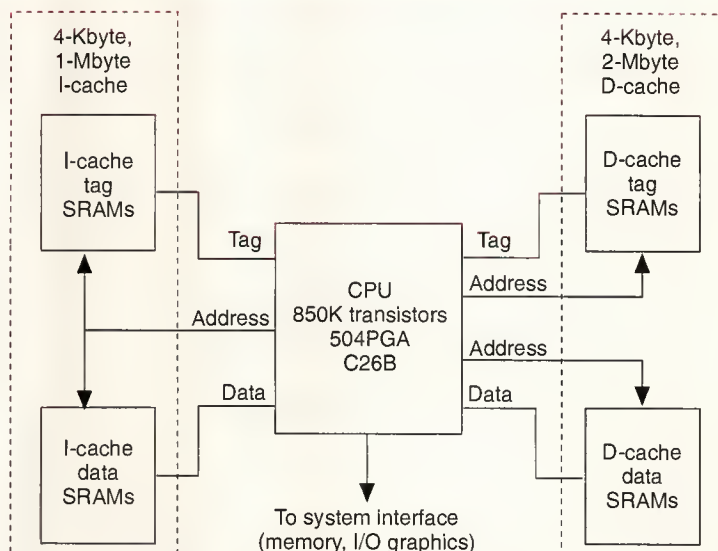


Figure 4. Processor block diagram.

Table 2. Cache configuration options and SRAM requirements.

Cache size (Kbytes)	Frequency (MHz)	SRAM size (Kbytes)	SRAM speed (ns)
256	50	32	15
256	66	32	12
1,024	96	128	9
256	99	32	9
256	132	32	7

Electrical design considerations. Although the external caches used by the PA7100 clearly have many advantages in performance, flexibility, CPU floor-space savings, and potential for future upgrades, their use did require a careful electrical design. Driving signals between chips at 100 MHz through printed circuit board traces and the chip's own package parasitic inductance and capacitance could have easily caused unacceptable delays. If we had not minimized these delays, the design would have required faster cache SRAMs or a reduced processor frequency, degrading the chip's performance. Requiring a faster SRAM would have quickly driven the cost of the cache too high, assuming an SRAM of the required speed was available at any price.

Complicating the design problem further was the high pin

***Timing control had to be flexible
to support a variety of
specifications for different types
of SRAMs, yet still make the
fullest use of the SRAM for
maximum processor frequency in
a variety of configurations.***

count required for the large buses between the PA7100 and its caches. Larger pin counts can increase the CPU package's footprint, which has the unfortunate effect of increasing the parasitic inductance and capacitance of the package. Besides increasing delays, high pin counts can also cause skews between the cache signals when some signals are affected more than others. Variations in the length of their interconnections and package load effects can generate such differences on the signals. Skew, like delay, can also lower the processor's frequency because it can limit the ability of a design to compensate for delay without decreasing the clock frequency.

To avoid the potential limitations these challenges might impose, the PA7100 design dealt with these issues directly. We tuned the PA7100 design to optimize the cache memory paths to get the highest processor frequency for a given speed of cache SRAM.

We decided to combine the floating-point unit with the integer processor unit into a single CPU chip for a variety of reasons, only some having to do with cache issues. Nevertheless, the cache design benefited because some of the signal pins that had previously been used for processor-to-coprocessor communication in the last PA-RISC implementation could now be used for CPU/cache signals instead. Even so, the CPU pin count increased by nearly 100 pins over its predecessor. We designed a new interstitial pin-grid array (PGA) package that not only accommodated the added pin count but even had a smaller footprint than PGA packages used by earlier PA-RISC designs. This actually improved the electrical performance of the package.

A single-chip processor configuration also cut the capacitive load on the instruction and data buses by more than half. Because these signals could now run directly between the CPU and the cache SRAMs and did not need routing to a third chip, the cache and CPU could fit much more closely together. This had several advantages. The design could match the signal lengths more evenly and keep them very short,

eliminating the need for 189 electrical terminations and further minimizing delays and skews. If the signal loading had not been reduced and the terminations had been required to keep the electrical characteristics acceptable, routing of signal traces would have been longer and more complex. This would not only have increased delays and skews but also would have wasted a large amount of power and board space. Our design also saved the cost of the components for 189 signal terminations.

Another major part of the design was controlling the timing of signal transitions. The timing control had to be flexible to support a variety of specifications for different types of SRAMs, yet still make the fullest use of the SRAM for maximum processor frequency in a variety of configurations. The signals needed to transition early enough that the receiving chip, either the CPU or SRAM, had adequate time to act on the signal but not early enough to interrupt the chip's previous action or miss the last signal value. To control this timing, we used special clocks to signal at what time the driver circuits could force a transition or, in the case of a receiver circuit, could latch the signal value. These clocks also specified the point in time that either the CPU or the SRAM could drive its data lines. This arrangement avoids the situation where both the CPU and the SRAM attempt to drive the same line at the same time but to different voltage levels. We created the special clocks by using circuits that drive the CPU's internal clocks through printed circuit board traces to get delayed clocks. By varying the trace lengths, we could change the delays. We could then adjust the cache memory circuitry's timing for a variety of SRAM requirements without making changes to the CPU.

Instruction cache performance features. So that the PA7100 can issue and execute two instructions per cycle, not only must instructions be supplied at the same rate from cache but they must be supplied early enough for issuing to either the floating-point or integer unit without limiting the processor frequency. Early design efforts made it clear that the decode time could limit the processor frequency or require faster SRAMs. To keep the SRAM requirements as relaxed as possible, the design included dedicated predecoded bits for the instruction cache. This reduced the amount of decode required during instruction execution by effectively moving it to the time when the instructions are copied into the instruction cache. Because the predecode bits can directly steer the instructions to the proper execution unit, the design requires no special ordering for dual execution of two instructions to occur.

We also optimized the PA7100 to recover some of the time required to copy a cache line from memory into the instruction cache. By optimizing the execution of instructions, we allowed the processor to begin executing an instruction as soon as the instruction is returned from memory and before it is copied into cache. The PA7100 will continue to execute

instructions as they come from memory until the processor branches or its pipeline interlocks. Overlapping instruction execution and the manner of copy-in saves a maximum of six cycles per instruction-cache miss.

Data cache performance features. The PA7100 design improved the data cache's performance for both loads and stores to cache. Earlier PA-RISC processor implementations performed stores to the data cache in a three-cycle process. They first read the old tag and data, then verified a hit in the cache, while merging in the new data, and then finally wrote the merged data back to the cache. Stores now take two cycles. The store's tag read first overlaps with the previous store operation. Then it is used to verify that the line targeted by the store is a hit, or present in the cache, before the write begins. See Figure 5.

Overlapping the tag read of one address while data is written to a different address required separate address buses for the cache tag and data SRAMs. The dirty bit—usually associated with the tag and used to indicate whether or not a line is dirty and needs to be copied back to memory—remained with the data access. The dirty bit has to be marked for copy back to memory with the data store, and the tag address has already changed for the next load or store operation by the time the write begins, making this arrangement necessary.

Completing the store in two cycles also meant that we had to eliminate the data read/merge portion of a three-cycle read/merge/write store. For single-word stores, we used separate write controls to the cache data SRAMs for each of the two data words. This way only the data to be changed is written, and it need not be merged with the rest of the double word written to the data cache. To limit complexity, this was done only for word stores, as they are the great majority of stores. Much rarer byte and half-word stores still must be merged in the CPU and suffer a pipeline penalty.

To further enhance the performance of store operations, the design included a new encoding of the store instruction. It provides a hint to the hardware that if the stored data's cache line is not already in the cache, copying the cache line in from memory is unnecessary. The software can use this feature for block operations such as block moves, zeroing large memory spaces, and block copies. In these instances, the whole line will be changed anyway and fetching the old data from memory would be a waste of time. In such a case, unless the line it displaces in cache is marked dirty, only writing the tag for the new line into cache is needed. If the displaced cache line is dirty, it is first copied out to memory after which the new tag is written. This feature not only greatly increases the processor's performance for this kind of code, but also decreases the traffic on the memory bus, thus improving the performance of multiprocessor systems.

Two new PA7100 features helped to improve processor performance during a data cache miss. A data cache miss arises when a piece of data is not found in the cache and

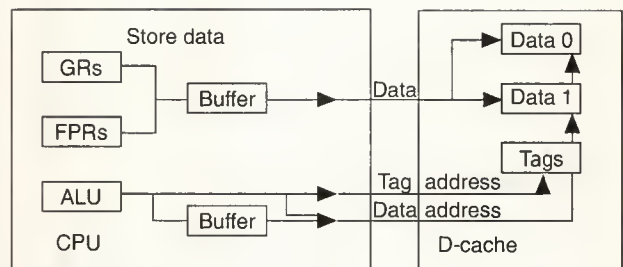


Figure 5. Store data path.

must have its corresponding cache line retrieved from memory. The "hit under miss" feature allows execution to continue even after a load or store has missed in the cache. Execution can continue until another cache miss occurs or the data from memory is required to complete execution of another instruction. Load-and-store operations can execute to lines in cache without stalling the CPU. During a store miss, word or double-word stores to the same cache line can execute. These options give software a great deal of flexibility in scheduling events for minimizing the penalty of a cache miss.

The design also included data cache "streaming" to allow execution to proceed as soon as possible. When the operand of an instruction is the target of an earlier load that missed in the cache, this feature allows the instruction to execute as soon as the critical word is returned from memory without waiting for the miss to complete.

The final data cache performance improvement involved the load-and-clear semaphore operation.⁹ Under certain circumstances the operation can complete in the processor's cache, reducing its pipeline penalty to that of any store instruction and reducing traffic on the memory bus.

Virtual address translation (TLB)

We have optimized the process of translating the PA7100's 48-bit virtual addresses to real addresses in hardware to support translations handled by both hardware and software. The translation look-aside buffer, TLB, is a fully associative first-level hardware unified TLB (UTLB). It has 120 fixed 4-Kbyte page entries and 16 variable size entries, each of which can map 512-Kbyte to 64-Mbyte spaces. Any entry can be "locked in" by software to keep the entry from being replaced by translations for different virtual pages. To avoid conflicts between instruction and data accesses in the UTLB, the design includes a single-entry look-aside buffer for instruction translations. Each TLB entry contains a 36-bit virtual page number and its comparator, a 20-bit real page number (RPN), a 22-bit protection vector, and a valid bit. The data TLB entries also contain three debug and trap enable bits.

In addition to the hardware TLB, software maintains a vari-

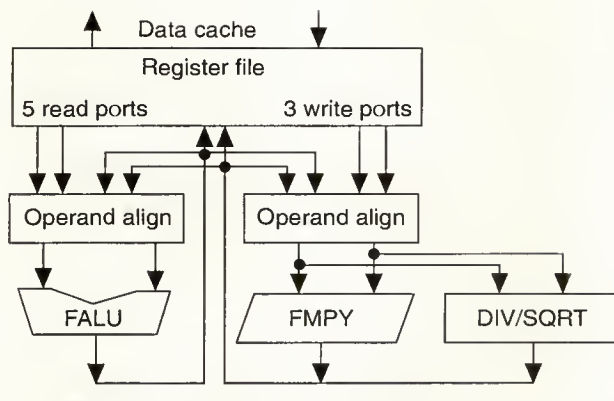


Figure 6. Floating-point block diagram.

able size, second-level TLB in memory that is read by a *hardware table walker* when there is a first-level miss in the hardware TLB. The physical page directory or PDIR contains this table.

We designed the hardware table walker (TLB miss handler) to reduce the TLB miss penalty, while keeping the PA7100 chip's area and complexity low. Encountering a first-level TLB miss, the PA7100 calculates the entry's address in the PDIR from the space register value and the virtual page number that missed in the UTLB. With the address determined, checking the PDIR entry for a valid matching tag begins. If there is a match, the chip inserts the real page number and protection information from the PDIR into the hardware UTLB and retranslates the original access. If the PDIR entry is not valid or does not match, its address passes to software and a trap is taken to the software TLB handler. Along with two other features this helps to minimize the software TLB miss penalty.

The general registers used in TLB software miss handling are automatically stored in "shadow" general registers as the trap is taken. The values are restored at the end of the trap handler with a special return-from-interrupt instruction that restores the corresponding general registers from their shadow registers. New, fast TLB insertion instructions also help reduce the software handler's miss penalty. Using these optimizing features reduces TLB miss handling delays significantly. There is no penalty for a first-level hit in the UTLB, and the penalty for a miss is as little as ten cycles if the table entry is in cache.

By redesigning the TLB hardware, we took advantage of special opportunities to enhance TLB performance. Because large segments of memory can be mapped off and locked in the translation entry, the new design improves operating system and graphics software performance by keeping TLB misses low for large pieces of operating-system code, tables, and graphics frame buffers. The single-entry instruction look-aside

Table 3. Floating-point instruction timing.

	Latency/dispatch (cycles)	
	Single precision	Double precision
ALU	2/1	2/1
Multiply	2/1	2/1
Multiply/ALU	2/1	2/1
Divide	8/8	15/15
Square root	8/8	15/15

buffer helps avoid contention with data accesses in the UTLB. Overlapping the buffer's update with the branch penalty also typically avoids its replacement penalty. The penalty for instruction TLB misses is, therefore, almost negligible when the entry is in the UTLB.

The PA7100 floating-point unit

The PA7100 floating-point unit contains five major subunits: floating-point pipeline control logic, a 32×64 register file, a floating-point arithmetic logic unit (FALU), a floating-point multiplier (FMPY), and a floating-point divide/square root unit (DIV/SQRT). See Figure 6. The floating-point data path implements IEEE 754 compliant single- and double-precision math. The floating-point unit provides exceptional floating-point performance for both technical and business computer systems. It achieves a peak execution rate of 200 Mflops at 100 MHz.

All floating-point operations except divide and square root (DIV/SQRT) are fully pipelined with a two-cycle latency for both single- and double-precision operands. The processor can issue an independent floating-point operation every cycle with no penalty cycles. Consecutive flops with a register dependency will incur a one-cycle penalty. Divides and square roots take 8 cycles in single-precision and 15 cycles in double-precision modes. Divides and square roots execute outside of the normal pipeline so that instruction execution does not stop until a dependency on the result register arises or another divide and square root is issued. This allows FALU and multiply instructions to execute in parallel with a divide or square root operation. Table 3 summarizes the timing for floating-point instructions.

Circuit density was a prime concern for the floating-point unit. Early in the design we saw that the highly parallelized algorithms commonly used in stand-alone coprocessor chip designs could not be compressed onto the CPU die. Furthermore, we realized that we would need fully combinatorial algorithms for the FALU and multiply circuits to achieve the

required level of performance. Using dynamic logic to exploit the well-known speed and density characteristics of that circuit type solved the problem. Typical dynamic circuits cannot perform inverted logic operations without introducing race hazards. To overcome this hurdle, we devised a system of self-timed logic. Using this method allowed us to design a multiplier and ALU that can compute full double-precision results in 20 ns.^{10,11}

Floating-point register file. The floating-point register file contains thirty-two 64-bit floating-point registers. Registers 0-3 contain the status register and exception registers. Registers 4-31 serve as operands for the floating-point units. In addition, FR0 is hard-coded to floating point 0 when used as an operand. The floating-point instruction set can access each register as a 64-bit double word or as two 32-bit single words. The register file has three write ports and five read ports, which allows concurrent execution of a multiply, an add, and a load or store operation.

The floating-point instruction set includes instructions that perform more than one floating-point operation. Multiple-operation instructions are five-operand instructions that combine a three-operand multiply with a two-operand add or subtract. The format of the multiple-operation instructions is:

FMPYADD RM1, RM2, TM, RA, TA.

The operands specified by RM1 and RM2 are multiplied, and the result goes into the register specified by TM. The RA and TA fields specify the source operands for the FALU operation. The result of the FALU operation goes into the floating-point register specified by TA. The multiported register file allows for the simultaneous launch and completion of both the multiply and FALU operations.

The design provides extensive register bypass capability that reduces penalties for floating-point operations with register dependencies. We provided paths to bypass load data and floating-point operation results as operands to the floating-point units without first going through the register file. Also, a store bypass path bypasses floating-point operation results to the cache interface. A floating-point operation followed by a store of the target register incurs no penalty, even if these two instructions are simultaneously dispatched.

Floating-point ALU. The floating-point ALU performs add/subtract, compare/complement, and convert instructions for

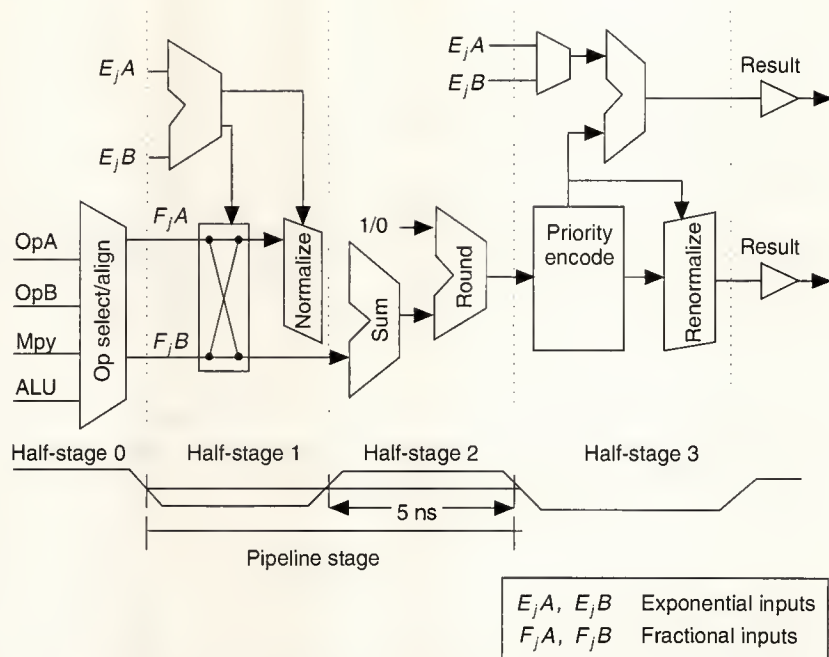


Figure 7. Floating-point ALU organization.

both single- and double-precision operands. The unit deviates from traditional implementations in that it performs floating-point additions, subtractions, and floating to/from integer conversions within a single functional unit. Traditional implementations perform additions and subtractions in one unit and conversions in another unit.

Shown in Figure 7, the floating-point ALU has four half-stages that correspond to the two states within the two-cycle latency. The first half stage, half-stage 0, latches the operands from the register file and checks for zero operands. Half-stage 1 shifts the significand of the smaller floating-point number to align the binary point. For a subtract operation, the smaller significand is complemented. Because the significands are unsigned, the design subtracts the smaller significand from the larger to avoid an extra complement operation in the case of a negative result. Half-stage 2 contains a 52-bit adder with rounding logic. Half-stage 3 contains a leading-one detector and a left shifter to postnormalize the result. Finally, the ALU drives the result back to the register file and optionally bypasses it to the operand buses or the cache store port.²

Integer to floating-point conversion involves normalizing the integer so that it consists of a significand in the form 1.xxxx with an appropriate exponent. This operation resembles the prenormalization step of addition or subtraction. The position of the most significant digit of the integer may be greater, however, than the number of bits in the destina-

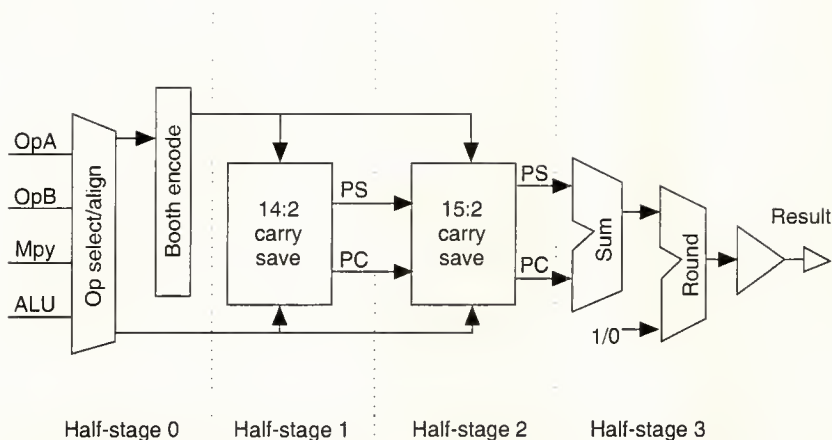


Figure 8. Floating-point multiplier organization.

tion significand. In this case, we must shift the integer to the right until it just fits into the most significant bit of the significand. Rounding is necessary if right shifting occurs. The additional hardware required beyond addition and subtraction is an integer leading-one detector to determine the amount of right shift.

Floating-point to integer conversions involve right shifting the significand of the floating-point number by the difference between the exponent and the exponent bias for the particular floating-point precision. Rounding becomes necessary if the right shift operation results in any lost digits. Having the generated integer in two's-complement form complicates the rounding process because the original significand is in sign-magnitude form. No additional hardware is required beyond that needed for addition.

Double-precision to single-precision floating-point conversion consists of right shifting the significand of the input operand 29 bit positions and rounding. Single-precision to double-precision floating-point conversion is a simple renormalization as is done for subtraction. Copy and absolute value operations require only multiplexers to add zero and modify the sign bit.

Because the hardware requirements for these operations are similar, the PA7100 floating-point ALU combines all of these operations into one functional unit. The additional hardware required beyond the traditional floating-point add/subtract unit is an integer leading-one detector, several multiplexers for the various operations, and additional control logic. This combined hardware approach saves significant area compared with traditional implementations.

Floating-point multiplier. The floating-point multiplier performs multiplies of single- and double-precision floating-point operands. In addition, integer multiplies of 32-bit unsigned integers provide a 64-bit result. Figure 8 diagrams the

floating-point multiplier organization.

There are four half stages in the multiplier pipeline that correspond to the two states within the two-cycle latency. The first half stage, half-stage 0, provides both encoding of one of the significands and the start of the exponent add and rebias. Half-stage 1 completes the exponent add operations and performs half of the partial product summation (PS). Half-stage 2 completes the partial product summation. The carry-propagate addition to generate the significand product, rounding, and renormalization occur in half-stage 3. Finally, the floating-point multiplier drives the result back to the register file and optionally bypasses it to the operand buses or the cache store port.

The largest portion of the multiplier is the array that performs the generation and summation of partial products. Although the Wallace tree is generally accepted as the highest performance multiplier array structure, we made some concessions in the PA7100 array on behalf of silicon area constraints. The performance advantage is regained by the use of a high-speed dynamic full adder circuit that provides a summation delay as low as 350 ps.

IEEE rounding imposes one twist on the significand logic that mandated the development of a unique carry-propagate adder to obtain a compact, fast solution. For correct rounding, the design may need to increment the final significand. We designed a carry-select adder architecture in the multiplier rounding logic. This adder splits the word into delay-balanced sections. Within each section, the design implements two carry chains: one assuming the carry into the section is a one, and the other assuming the carry-in is zero. A second level of carry logic propagates the carry to the most significant bit position of the next section. This and other information determines the correct sum that can be rapidly produced by multiplexing the correct carry chain in each section to the single-gate-delay sum generator. In this way, the multiplier can generate the correct sum without duplicating the entire adder and multiplexing the correct answer. Also, the duplicate carry chain is part of the speed-enhancing multilevel carry scheme, not just overhead to generate the correct sum.

Floating-point divider. The divide/square root unit performs single- and double-precision operations. As diagrammed in Figure 9, we implemented this unit as a separate block that allows multiplies to continue even while a divide is in progress. The iterative divide/square root unit uses a modified radix-4 SRT algorithm that is a nonrestoring digit-by-digit method and is essentially an adaptation of hand division. (SRT division is a nonrestoring division algorithm—named for its developers, Sweeney, Robertson, and Tocher—that

uses a redundant-digit set.) Digit-by-digit division can be very slow, but by increasing the radix, each digit represents multiple bits of the quotient. The total number of iterations can then be reduced. As the radix increases, the complexity of SRT division grows exponentially.

Because of the simplicity of the radix-4 division hardware, the circuits can run at twice the system clock rate, achieving effective radix-16 performance with the low hardware cost of radix-4. The unit computes two bits of the quotient on each iteration. With two iterations during each clock cycle, four quotient bits are computed each clock cycle. With the divider running at twice the main clock frequency, the performance of the unit compares to high-performance Newton-Raphson dividers and requires only a fraction of the hardware cost.

Hardware underflow mode. The PA7100 implements a quick hardware underflow mode. This mode, which is enabled by setting a bit of the status register, eliminates the overhead of a trap handler when denormalized operands are present or when the result of an operation underflows. In the hardware underflow mode, operations that would normally signal the underflow exception return a zero result with no exception. In this mode, the PA7100 treats input denorms as signed zeroes. It detects the inexact flag and inexact exception just as in the IEEE mode except that it treats denormalized operands as signed zeroes. When a result is flushed to zero it sets the inexact flag. Note that when this mode is enabled, computations do not comply with the IEEE floating-point standard.

Memory and I/O interface

The PA7100 has a system interface bus (named P-bus) that services cache misses and I/O transactions. Although it is only 32-bits wide, the P-bus can operate at the pipeline clock frequency. In addition to the 32 address and data lines, the P-bus uses 17 protocol signals to allow data to flow on the P-bus at a rate near the bandwidth limit. The P-bus protocol includes cache-line-sized (32-byte) transactions, split read/return transactions, and single-cycle read requests. The P-bus also includes TLB and cache coherency transactions for supporting multiprocessor configurations. Because graphics applications are important for many systems that use the PA7100, we optimized the PA7100 pipeline and P-bus interface to sustain an I/O write bandwidth equal to one half of the P-bus bandwidth.

In a system design using the PA7100, a processor memory interface (PMI) connects the P-bus with the memory and I/O subsystems. Figure 10 illustrates a uniprocessor configuration. In workstation applications, we optimize the PMI to be a low-latency controller that is tightly coupled to the memory DRAM arrays and the I/O bus. In high-end commercial and technical server applications, the PMI is a bus converter that connects the P-bus to a higher-bandwidth system bus.

The earlier 66-MHz PA-RISC CPU used the P-bus as a system interface bus. We wanted to maintain compatibility with

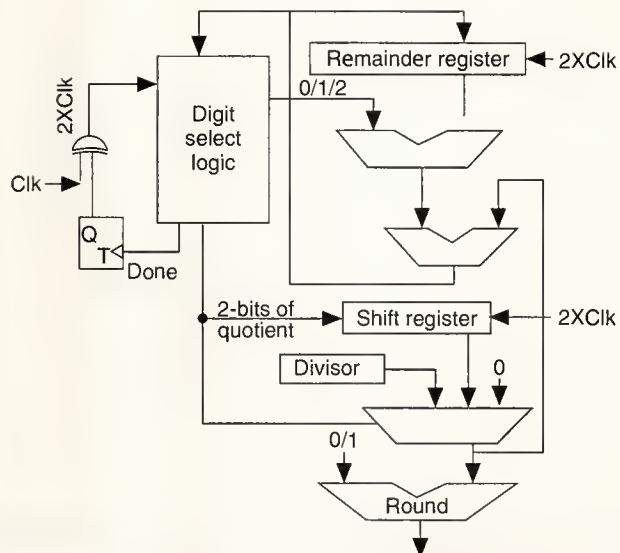


Figure 9. Floating-point divider unit.

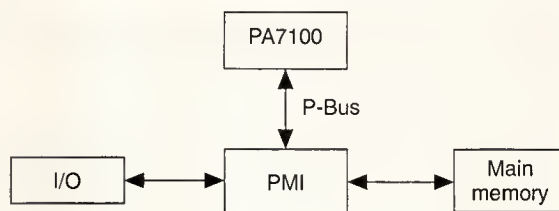


Figure 10. Uniprocessor configuration.

that system bus so that the 100-MHz PA7100 could serve as a processor upgrade for previous Hewlett-Packard computers that used a 66-MHz P-bus PMI. Buffering functionality, implemented in the P-bus interface, allows three pipeline-clock-to-P-bus-clock frequency ratios. These ratios are 1:1, 3:2, and 2:1. The 3:2 ratio allows the PA7100 to serve as a 99-MHz processor upgrade in the earlier 66-MHz systems. In addition to simple frequency conversion, the buffering control makes better use of the slower P-bus bandwidth by packing transactions more densely than would have occurred at the full-frequency, 1:1 ratio. The design accomplished this by eliminating wait states and by overlapping transactions.

Multiprocessing. The PA7100 includes hardware support for implementing shared-memory multiprocessor systems. The PA7100 implements the PA-RISC instructions for purging and flushing the caches and TLB.⁹ The PA7100 supports two basic system configurations for constructing multiprocessor systems. One configuration is suitable for scalable, high-end systems, and the other makes possible a low-cost, dual-

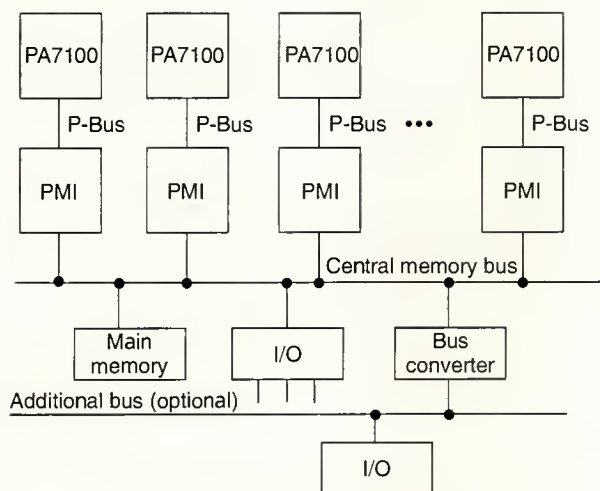


Figure 11. Scalable multiprocessor configuration.

processor system.

The PA7100 implements a write-back data cache policy. Each D-cache line can exist in one of four states:

- 1) invalid,
- 2) private-clean, meaning that the line is valid exclusively in the data cache of one processor and has not been modified with respect to the copy in main memory,
- 3) private-dirty, meaning that the line is valid exclusively in the data cache of one processor and has been modified but not yet posted to memory, and
- 4) shared, meaning that the line is unmodified and may be valid in more than one data cache.

The shared state is never used in a uniprocessor.

A coherent PA-RISC multiprocessor system using the PA7100 must behave as if there were logically a single data cache, a single instruction cache, and a single TLB. Cache flushes, cache purges, and TLB purges executed on one processor are broadcast to all other processors in the system. Hardware must maintain data cache coherency automatically. When a data-cache miss occurs, hardware will perform a cache coherence check by interrogating all other data caches in the system for the current data. The instruction cache is read-only, and instruction references need not be satisfied by a cache coherence check; software is responsible for modifications to the code stream.

Scalable high-end system. A scalable, high-end, shared-memory multiprocessor system using the PA7100 would be organized as shown in Figure 11. In this configuration, the PMI is responsible for maintaining cache and TLB coherency. Each PMI snoops on the shared bus, watching for co-

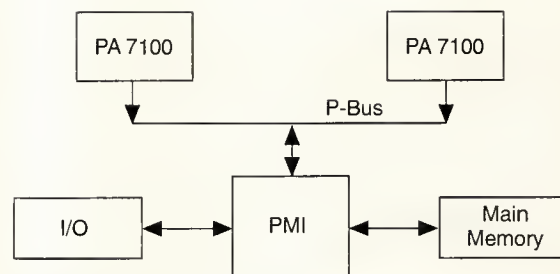


Figure 12. Dual-processor configuration.

herent read, flush, and purge transactions issued by other modules on the bus. If a coherent transaction occurs, the PMI will initiate a coherency check or issue a flush or purge transaction to its CPU via the private P-bus connection. The PA7100 CPU will perform the requested cache or TLB action. If appropriate for the transaction type, the CPU then possibly will write back a private-dirty line over the P-bus to the PMI. When a dirty line is written back, the PMI will write the line on the shared bus so the line can be posted to memory, forwarded to another PMI-CPU pair that requested the line, or both. Sufficient information is available on the P-bus for the PMI to maintain a set of cache tags that mirrors the tags in the CPU cache. Coherent transactions therefore need to be forwarded to a CPU only when a target line is actually present in the cache.

This PMI-per-processor, shared-bus multiprocessor organization yields performance that scales well as the number of processors is increased. Implementing this organization is relatively expensive because the component count is high. Also, the bandwidth required to support multiple PA7100 processors requires an electrically sophisticated backplane and a highly interleaved memory controller design. This organization is appropriate for high-end server applications using the PA7100 in which the processing throughput that can be achieved justifies the cost of the implementation.

Low-cost dual-processor system. The PA7100 includes cache and TLB coherency functionality for implementing a low-cost, dual-processor system. Figure 12 illustrates the organization of this system. This organization does not require a PMI that implements the coherency functionality described for the scalable high-end system; the PA7100 processors carry all of the burden for maintaining coherency. The same low-latency PMI with tight coupling to DRAM and I/O that is used in a uniprocessor organization can be used for a dual-processor system. This organization supports the three pipeline-to-P-bus frequency ratios.

Essentially, the two processors share a single P-bus and appear to the PMI as a single, particularly bandwidth-hungry PA7100 CPU. Each CPU watches the transactions issued by

the other and internally performs the appropriate coherency transactions. When one processor issues a transaction that requires the other to write back a dirty line to memory, the first processor automatically, and transparently to the PMI, relinquishes control of the P-bus to the other. When one processor issues a read transaction for a line that is in the other processor's cache in the dirty-private state, the line is immediately transferred across the P-bus into the other cache. The line is not first posted to memory, and the requesting processor does not need to retry the read transaction.

Wanting to take advantage of the low-latency service the uniprocessor PMI can provide and desiring to maximize the use of the bandwidth available on the single P-bus, we paid particular attention to the arbitration protocol that the processors use when contending for the P-bus. The arbitration penalty is at most one state. If one processor wants to issue a transaction after having issued the previous transaction, and if the other processor is not simultaneously contending for the P-bus, the one processor may immediately issue its transaction without paying an additional arbitration penalty state. This feature allows a processor to issue a burst of transactions after using an arbitration state only for the first transaction. If one processor wants to issue a transaction after having not issued the previous transaction, it may do so after a one-state arbitration penalty. The arbitration protocol is fair in that each processor wishing to issue a P-bus transaction will wait for at most one of the other processor's transactions to complete before gaining control of the P-bus.

PA7100 methodologies

Bringing the PA7100 chip to fruition required a series of design, test, and verification methodologies. These methodologies allowed for a faster design cycle and greater quality of final product.

Design methodologies. Many of the choices made during the design phase of the PA7100 greatly influenced the performance and time to market of this processor. The PA7100 is composed of custom and semicustom block designs. To achieve maximum performance, we made critical circuits such as the TLB and IO structures from custom layouts. Since we leveraged many of the custom circuits on PA7100 from previous designs, our designers did not incur long development times. We made other circuits such as the data path and control logic from semicustom libraries that allowed for quick layout and easy design changes.

Our design places extra gates in and around major control blocks on the PA7100 CPU that could be used for the repair of functional and electrical defects. Using these spare gates for mending flaws meant that only metal layers needed to be changed, allowing for quicker turnaround of the silicon. Since some defects can mask other defects, decreasing the time it takes to evaluate bug fixes allows us to find other problems more quickly.

Since we leveraged many of the custom circuits on the PA7100 from previous designs, our designers did not incur long development times.

Test methodologies. To achieve the highest performance systems possible from the PA7100, testing for the part must be of the highest quality and allow for accurate binning. To attain this goal, the testing for the PA7100 employed a two-pronged approach: parallel-pin testing and serial-scan testing.

Serial testing, performed through a diagnostic port, allowed us to subdivide each PA7100 into blocks for thorough proof. The visibility provided by the large number of scan latches improved the process of finding processing defects. The scan latches also allowed for complete control of internal blocks, which permitted compact and effective vectors for finding faults.

Parallel-pin testing permitted an accurate determination of PA7100 speed and functional coverage of data blocks such as general registers, TLBs, and floating-point units. These blocks do not require complicated vector sequences for effective coverage. We generated parallel-pin vectors for the PA7100 from compiled code, allowing any code sequence found effective for stressing part speed to be used in the production test. This direct port from failing code sequences to test screen ensured the highest quality of speed binning.

Verification methodologies. We designed the verification techniques used for the PA7100 CPU, both before and after silicon, to bring out problems as quickly as possible. To ensure that first silicon would work, we used a description-level simulator with the descriptions employed at the lowest hierarchical level possible. We used schematic representations for all higher levels. This simulator could run code of any type, and we used an emulation program to check state-by-state results. Using this method meant that we were not restricted to self-checking code.

After silicon was produced, we checked the quality of design using many techniques. One new technique—using two types of pseudorandom code generators developed for the PA7100—proved very useful. We used the first type on the floating-point coprocessor. Floating-point emulation is a built-in function of Hewlett-Packard PA machines and is the check used by the random floating-point code generator. The code produced by this generator originated from a template that restricts the resulting code to certain instructions and sequences

Table 4. PA7100 workstation performance.

System	Frequency (MHz)	Cache (I/D) (Kbytes)	Memory (Mbytes)	SPECint 92	SPECfp 92	Dhrystone (MIPS)	SPECmark 89	SPECint 89	SPECfp 89	Linpack Dbl 100x100
HP9000/735	99	256/256	64	80.0	150.6	124	146.8	88.1	206.2	40.8
HP9000/715-50	50	64/64	32	36.5	72.1	62	69.0	40.2	99.0	13.2
HP9000/715-33	33	64/64	16	24.2	45.0	41	45.9	26.8	65.6	8.9

Table 5. PA7100 commercial system performance.

System	Frequency (MHz)	Cache (I/D) (Kbytes)	Memory (Mbytes)	TPC-A host- based	TPC-A client- server	AIM perf. index	AIM max. user load	AIM jobs/ minute	NHFS- stones
HP9000/G50	96	256/256	64	153	184	69.2	620	677	1,320 @39

of instructions. Its design involved comparing random code sequences that were executed against emulation of the same sequences. If the results differed, we saved the sequence for later debugging.

The second random code generator had a wider focus, and almost any instruction could be tested using this technique. We ran the template for this generator on a simulator and recorded the check sums. When the generator ran on the template, it would randomly change the environment under which the code was running. This was done by randomly inserting traps, cache misses, and modifying the mode of the PA7100 performance options. Once again, if a wrong check sum arose, we would log the failing code for later debugging. At the speed the PA7100 runs, random code can quickly squeeze out defects in the design.

Once we had discovered failing code sequences and determined that the cause of the failures was not readily reproducible in the simulators, we could debug the failure by using a scan path dumper. Many types of problems could not be reproduced in simulators, including speed paths, race conditions, or any electrical defect. We could compile any code and run it on the tester in the parallel-pins mode. Since the tester has full control over all chip pins, it can stop clocks at any time. With the clocks stopped at a predetermined state, the tester can interface with the PA7100's serial scan port and scan out the entire internal state of the chip. Doing this for consecutive states built up a log of the internal state of the part. We

then compared this log with a log made by repeating the scan path dump at a passing frequency. If there were no passing points, we could compare the log to one produced from the description-level simulator. This technique prevented any problem from remaining misunderstood for very long.

THE PA7100 CPU IS CURRENTLY AVAILABLE in midrange commercial multi-user systems as well as a range of desktop workstations. Tables 4 and 5 give measured values for popular benchmarks. Hewlett-Packard is currently shipping these products in volume to its customers. ■

References

1. D. Tanksalvala et al., "A 90-MHz CMOS RISC CPU Designed for Sustained Performance," *Digest of Technical Papers, IEEE Solid-State Circuits Conf.*, Vol. 33, Piscataway, NJ, 1990, pp. 52-53.
2. C. Gleason et al., "CMOS Processor Circuit Design in Hewlett-Packard's Series 700 Workstations," *Proc. Int'l Conf. Computer Design*, IEEE Computer Society Press, Los Alamitos, Calif., 1991, pp. 288-292.
3. W. Jaffe et al., "A 200-Mflop PA-RISC Processor," *Proc. Hot Chips Symp. IV*, IEEE Computer Society Technical Committee on Microprocessors and Microcomputers, 1992, pp. 1.2.1-1.2.12.
4. E. DeLano et al., "A High Speed Superscalar PA-RISC Processor,"

Digest of Papers, Compcon, CS Press, 1992, pp. 116-121.

5. R.R. Oehler and M.W. Blasgen, "IBM RISC System/6000: Architecture and Performance," *IEEE Micro*, Vol. 11, No. 3, June 1991, pp. 14-17, 56-62.
6. S. Mirapuri, M. Woodcare, and N. Vasseghi, "The Mips R4000 Processor," *IEEE Micro*, Vol. 12, No. 2, Apr. 1992, pp. 10-22.
7. K. Diefendorf and M. Allen, "Organization of the Motorola 88110 Superscalar RISC Microprocessor," *IEEE Micro*, Vol. 12, No. 2, Apr. 1992, pp. 40-63.
8. R. Lee, "Pathlength Reduction Features in the PA-RISC Architecture," *Digest of Papers, Compcon*, CS Press, 1992.
9. R. Lee, "Precision Architecture," *Computer*, Vol. 22, No. 1, Jan. 1989, pp. 78-91.
10. C. Heikes et al., "The Design of 200-Mflop Floating Point Megacells for the PA7100," *Proc. Design Technology Conf.*, Hewlett-Packard, San Diego, Calif., May 1992, pp. 25-32.
11. J. Yetter, "A 100-MHz Superscalar PA-RISC CPU/Coprocessor Chip," *Digest of Technical Papers, Symp. VLSI Circuits*, 1992, pp. 12-13.



Tom Asprey is a member of the technical staff at Hewlett-Packard's Fort Collins site where he has been involved with CPU, cache controller, and system I/O chip designs for PA-RISC designs. He received a BS degree in electrical engineering from New Mexico State University. He is a member of the IEEE and the IEEE Computer Society.



Gregory S. Averill is a member of the technical staff at Hewlett-Packard where he has been involved in VLSI chip design and multiprocessor system design. He received a BS in electrical engineering from Brigham Young University and an MS in electrical engineering from Stanford University. He is a member of the IEEE and the IEEE Computer Society.



Eric DeLano is a member of the technical staff at Hewlett-Packard in Fort Collins, where he has been involved in the control design and definition of five VLSI CPU designs for the PA-RISC architecture. He received both BS and MEng degrees from the University of California at Berkeley.



Russ Mason, a member of the technical staff in Hewlett-Packard's Engineering Systems Lab, held responsibility for the design and verification of the PA7100 floating-point unit. He designed the floating-point register file and operand alignment circuits. In addition, Mason performed electrical verification of the PA7100 design. His current interests include the high performance circuit design for Hewlett-Packard's PA-RISC CPUs. He received a BSEE from Mississippi State University, Starkville, Mississippi.



Bill Weiner has participated in the design of high-performance PA-RISC computers as a member of the technical staff at Hewlett-Packard. He holds a BS in electrical engineering from Rensselaer Polytechnic Institute.



Jeff Yetter, as a member of the technical staff, has been engaged in the development and management of integrated RISC processors with Hewlett-Packard's System Technology Division. He holds a BS in computer engineering from the University of Illinois at Urbana.

Readers may direct questions concerning this article to Bill Weiner at Hewlett-Packard, 3404 E. Hamony Road, Fort Collins, CO 80525; or via e-mail at wnw@hpeswnw.fc.hp.com.

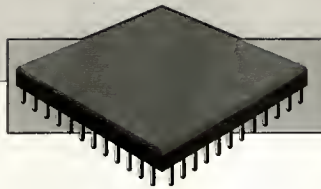
Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 153

Medium 154

High 155



The Alpha AXP Architecture and 21064 Processor

The Alpha AXP 64-bit architecture forms the basis for a series of high-performance computer systems. Building on almost 10 years of internal research into reduced-instruction-set computer architecture, Alpha AXP emphasizes performance and longevity. The 21064 microprocessor is the first Alpha AXP implementation. Operating at speeds up to 200 MHz, this chip serves as the heart for current systems that offer the highest microprocessor-based performance in the industry.

Edward McLellan

*Digital Equipment
Corporation*

The 64-bit Alpha AXP architecture^{1,2} and the first implementation DECchip 21064 microprocessor grew out of a multiyear effort at Digital. Our aim was to develop a computer family capable of leadership performance for the foreseeable future over a wide variety of applications. Combining strengths in semiconductor technology, computer architecture, hardware design, operating systems, compilers, and applications software, this effort recently delivered a series of such machines. Systems range from personal computers to workstations to supercomputers. Operating systems support includes OpenVMS, full 64-bit Unix (DEC OSF/1), Microsoft Windows NT, and soon, native Novell NetWare. Figure 1 shows the packaged DECchip 21064 microprocessor.

Our rich history of computer design spans 35 years and includes the 16-bit PDP-11 and 32-bit VAX computer families. The Alpha AXP architecture represents a new step in that evolution, one that combines full 64-bit address and data capabilities with principles of RISC architecture. Roots of the AXP development go back to the mid 1980s, when multiple investigations of RISC technology culminated in the definition of the internal Prism architecture. That definition included the valuable experience of completely designing a 32-bit microprocessor.³

In 1988, a task force chartered with exploring future enhancements to the VAX concluded that a new architecture would soon be necessary to extend the increasingly cramped 32-bit addressing space of the VAX.⁴ The Alpha AXP architecture went much further than that by addressing features such as multiple-instruction issue, multiple processors, and operating system independence. The Alpha AXP architecture benefits from the experience of a broad base of computer architects, hardware designers, and systems and applications software experts. The architecture strives to anticipate future trends as much as it attempts to provide current solutions. This architecture provides flexibility for both architectural evolution and hardware implementation over time in a variety of ways.

For any modern computer architecture to be successful, though, access to a strong semiconductor design and technology base is essential. Our semiconductor development began in the late 1970s with a double-metal NMOS process designed specifically to support high-performance microprocessors. Since then, our designers have developed four generations of CMOS technology. Each generation allows a straightforward path to shrink previous designs for advantages in speed, power, reliability, and cost. The 21064 microprocessor is designed in the CMOS-4 process, which offers 0.75-

μm feature sizes, three levels of aluminum interconnection, and a 3.3-volt power supply.

A new class of systems requires a tremendous amount of software development. Compatibility with older code was paramount for taking fullest advantage of the new architecture. The software task required more time than the hardware development. Therefore, we staged the hardware design to produce early development units that could assist software efforts. Prior to the final CMOS-4 version of the chip, we produced a CMOS-3 device that offered smaller caches and had no floating-point hardware support. This strategy allowed operating systems and internal developers to run code on actual AXP systems almost two years before the product shipment date. At the same time, the final hardware design got to take advantage of the latest semiconductor process advances.

Compatibility with a large, existing customer base of software also concerned us. Rather than burden the hardware with extensive support hooks, or restrict the architecture with compatibility issues, our design efforts adopted the idea of binary translation. Binary translation involves converting executable programs compiled for one hardware platform to another without requiring recompilation from original source code. For maximum reliability, the challenge also includes a runtime environment to support translation of almost all user mode applications and an interpreter to execute code that is not exposed by the initial translation. All of this must come together to produce a translated image that equals or exceeds the performance of the system being replaced. Translators for both VAX and Mips systems to AXP are available and have been invaluable in the highly successful migration of both system and user programs.

Alpha AXP architecture

Perhaps the most notable difference exhibited by the Alpha AXP is found not in a list of its features, but in its careful avoidance of quick-fix solutions to a variety of problems in computer design. Instead of a segmented address space, which can be more difficult to program, the design provides a large, 64-bit linear address space. Virtually all other computer manufacturers have 64-bit extensions planned, but only one other currently delivers 64-bit hardware. Only Alpha AXP offers a full 64-bit operating system with DEC OSF/1. A clean start rather than extension of a 32-bit architecture avoids hardware baggage that can include "orphan" 32-bit instructions (for example, 32-bit shifts) and other compatibility issues associated with old 32-bit software.

In Alpha AXP, all operations, including a small set for efficient 32-bit support, read and write full 64-bit quantities. To facilitate multiple-issue implementations, the architecture explicitly avoids condition codes, special registers, side effects, suppressed instructions, and branch delay slot instructions. These features fit well with single fetch and issue processors,



Figure 1. The packaged 21064 microprocessor.

but only complicate multiple-issue designs and often lead to performance bottlenecks. In a machine executing more than one instruction at a time, a single copy of any resource can become a point of contention. Likewise, a single skipped or forced instruction execution, as in the case of branch delay slots, does not fit well with the notion of a machine that fetches and executes multiple instructions each cycle.

The architecture also avoids direct hardware support for features that, although otherwise useful, are either uncommon or would likely limit the performance of anticipated systems through cycle-time restriction. Instead, the design provides support in a manner consistent with the architectural directions, but using software assistance for full functionality. Examples include the lack of direct-byte load/store instructions and precise arithmetic exceptions. A critical shift and multiplexer path is necessary for byte loads that can threaten cycle time. In addition, byte store operations require costly read-modify-write sequences in systems incorporating common error-correction code protection schemes. Byte writes with such ECC schemes can complicate and slow critical write-back cache designs. Recent experience has shown that some byte oriented codes run much faster when efficiently using the natural 64-bit (8-byte) data width and the byte manipulation support instructions provided.

Where byte operations are required, as in I/O support routines, designers of the first Alpha AXP PC have successfully used alternatives such as encoding sizes on address bits and encapsulating the byte manipulation code to port the Microsoft Windows NT operating system without changes to low-level driver code. In fact, byte manipulation encapsulation is identical to I/O operation encapsulation, which is necessary for using Intel X86 in and out instructions with high-level

***The Alpha AXP architecture is a
traditional RISC load-store
architecture—all data moves
between memory and registers
without computation.***

languages. A driver that already abstracts I/O operations need not be modified at all for use on Alpha AXP platforms.

As high-end processors such as Cray have done for years,⁵ hardware support for arithmetic traps is imprecise with respect to the instruction stream. An operator can choose precise trap behavior when necessary through the use of the trap barrier instruction, typically during program debugging. General use of the trap barrier, however, can allow precise arithmetic exception behavior at all times without appreciably degrading performance. Measured differences on the 21064 range from less than 1 percent in integer to between 3 and 25 percent in floating-point codes. Advantages in cycle time and design complexity allowed by this approach, however, compare favorably with these differences.

The Alpha AXP architecture is a traditional RISC load-store architecture. That is, all data moves between memory and registers without computation. Computation is done between data in general-purpose registers only.

Operating system independence. Anticipating the need to support multiple operating system ports, a set of privileged software subroutines, called PALcode, can tailor some of the lowest level hardware-related tasks unique to a particular operating system. For flexibility in service, interruptions, exceptions, context switching, memory management, and error handling all have controlled entry points in PALcode. Neither the hardware nor the operating system then is burdened with a bad interface match, and the architecture itself is not biased toward a particular computing style. In addition, since PALcode mediates all access to physical hardware resources, including physical main memory and memory-mapped I/O device registers, users can also tailor the code for special purpose environments such as real-time and highly secure computing.

Addressing. Virtual addresses are a full 64-bits wide, although subsets are allowed. The AXP employs little-endian byte addressing, similar to Intel X86 and VAX computers. Systems can access both big- and little-endian data using the byte manipulation instructions with a single instruction modification to the sequence. In fact, Digital and its partners are building both big- and little-endian systems and software.

Implementations may subset the address width, to a minimum of 43 bits with sign extension, but must check all 64 bits for compatibility with future systems. The AXP does virtual-to-physical-address mapping on a per-page basis, and its pages are 8 Kbytes with future expansion defined.

Data types. The fundamental unit of data is the 64-bit quadword, although the architecture also supports 32-bit longwords. Floating-point data types include both VAX and IEEE formats in both 32-bit single- and 64-bit double-precision formats. An extended-precision floating-point format is not included, but the designers have anticipated expansion by reserving a function field. Byte and word (16-bit) data types are not supported by direct load-and-store instructions but by short sequences of instructions. They can be manipulated in registers using normal arithmetic and the byte manipulation instructions.

Processor state. The hardware processor state includes separate 32-entry by 64-bit integer and floating-point register files. R31 is always zero in each file. Completing the required state are a longword-aligned, 64-bit program counter, floating-point control register for IEEE compliance, and a pair of lock registers for multiprocessor support. If the FETCH/FETCH_M instructions, or VAX-translated images are supported, additional hardware state is required.

The Privileged Architecture Library (PAL) gives designers the option of adding PAL state to the existing hardware state. The PALcode completes the architectural definition in an operating-system specific way. The hardware designers determine the implementation of PAL state, which can range from full hardware to full software or a combination of the two based on design constraints. Typical PALcode state include kernel stack pointer, user stack pointer, and translation look-aside buffers as well as a process-unique value for threads and a processor number for multiprocessor dispatch.

Instruction formats. As shown in Figure 2, the architecture uses four fundamental instruction formats: operate, memory, branch, and CALL_PAL. All instructions are 32-bits wide and contain zero to three register fields. To minimize register file port requirements, register B (RB) is never written and register C (RC) is never read.

The operate format includes arithmetic, logical, shift, and byte manipulation instructions. Scaled add/subtract and compare bytes instructions allow efficient operation on arrays and strings. Conditional move instructions for both integer and floating-point data, which test one input operand and optionally transfer data from another, remove branches in favor of a single instruction. Rather than using a single condition code location, the compare instructions write directly to any general-purpose register. They include an unsigned comparison operation for extended-precision arithmetic. There is no integer divide instruction. Where necessary, a 128-bit multiply can be used for emulation. The architecture enables traps on a per-instruction basis to avoid mode registers, and

provides some longword (32-bit) operations for compatibility.

Memory format instructions are mainly loads and stores, but also include some additional instructions. Loads and stores use two registers, specifying a base address and a data source or destination. The effective address calculation sign extends a 16-bit displacement to 64 bits and adds the 64-bit base register value. The architecture also provides load-and-store operations for longword (32-bit) quantities. General operations move aligned data quantities and trap on unaligned references, but instructions that mask the unaligned address bits and do not trap are available for use with the byte manipulation instructions. Calculated jump instructions also use the memory format; these instructions determine the target address directly from the base address without using the displacement field. The unused bits, however, are defined as hints for hardware prefetching mechanisms to improve pipeline efficiency. The additional hint information designates a likely target, allowing the hardware to continue fetching before the true target is available from the register file. If the hint is wrong, a misprediction restart costs no more time than if the hardware stalled waiting for the true address. A pair of load address instructions also use the memory format and allow a convenient way to create large constants using the 16-bit displacement field.

The design provides branch format instructions for both integer and floating-point data. These instructions test a single register for an operation-code-specified condition, and either branch to the target or fall through. To calculate targets, the instructions add a 21-bit longword displacement field to the updated program code (PC) resulting in a ± 4 Mbyte relative branch range. The large range effectively reduces the need for branches around or to other branches.

The CALL_PAL format instruction contains only a 6-bit operation code field and 26-bit function field. There are no explicit registers because individual instructions can be redefined for specific use. When executed, these instructions dispatch to PAL routines that perform an atomic, or uninterruptable sequence of instructions. CALL_PAL instructions then can serve to emulate complex instruction-set computer functionality.

Shared-memory multiprocessing. Scalable performance was an integral part of the architectural definition. Since cycle time and multi-issues for single processors are likely to become limiting factors over the lifetime of the architecture, multiprocessor support was critical to achieving both performance and longevity goals. The basic multiprocessor interlocking primitive for updating a shared-memory location is a RISC-style load-locked, in-register modify, store-conditional sequence of instructions. If the sequence completes without interruption, exception, or an interfering write from another processor, the store-conditional instruction succeeds and returns status indicating that an atomic update was performed. Otherwise, the store-conditional fails, and the program must branch back and retry the sequence. This mechanism scales

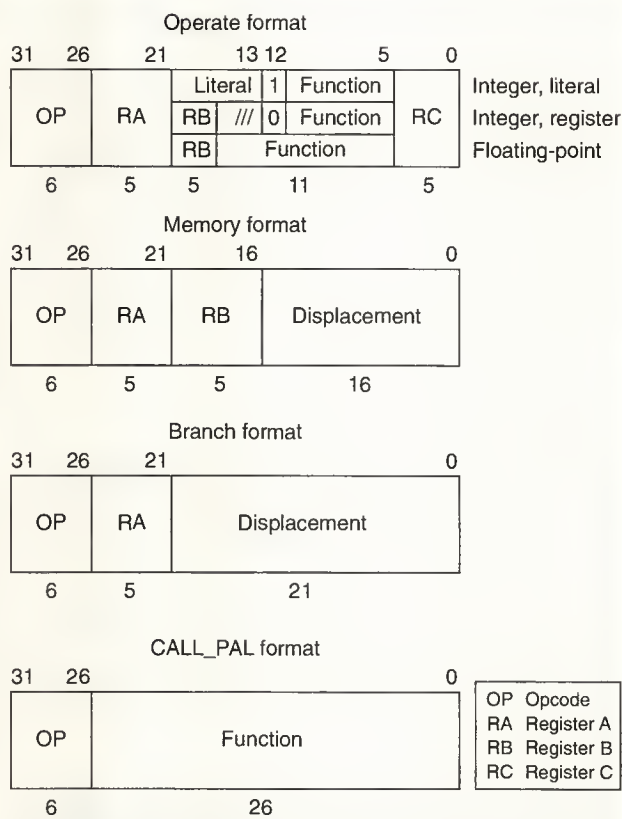


Figure 2. Instruction formats.

well with processor performance and allows multiple simultaneous noninterfering sequences.

The Alpha AXP architecture is the first RISC architecture to offer a relaxed, or weak memory-ordering model. SPARC V9 and PowerPC have more recently announced support for a weak-ordering model. Relaxed ordering implies that the sequence of reads and writes as viewed by another processor need not be in order. Multiprocessors that employ strict-ordering models are possible, but can be subject to performance limitations. For example, if a processor is designed to retry writes that result in errors, a strict-ordering model implies that the retry must complete before any other read or write occurs. This constraint excludes pipelined memory systems that would otherwise allow operations begun prior to the error to complete before, and out of order with the retry.

When strict ordering is required, as is the case in some I/O or multiprocessor synchronization operations, the Alpha AXP architecture specifies a memory barrier (MB) instruction to force serialization of operations. Software then controls serialization, enforcing it only when necessary. The lack of implicit ordering enables a variety of high-performance implementation techniques.

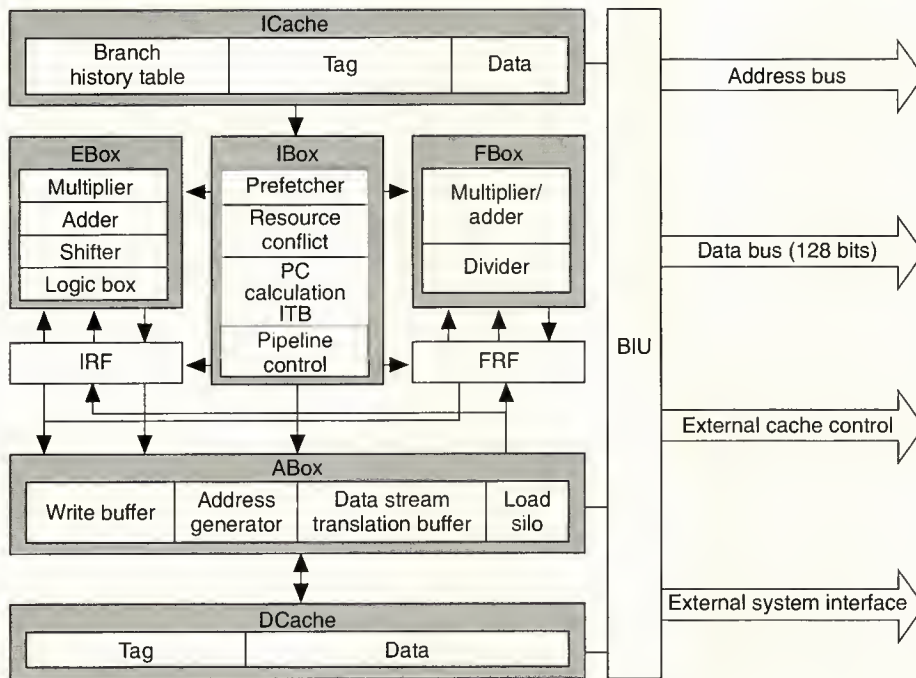


Figure 3. The 21064 block diagram.

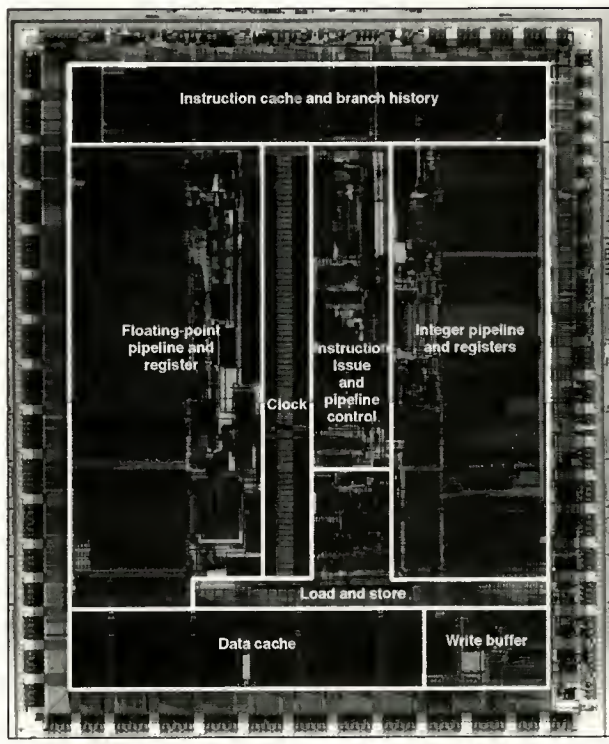


Figure 4. The 1.4 x 1.7-cm CMOS 21064 chip.

The 21064

The 21064 microprocessor is the first implementation of the Alpha AXP architecture.^{6,7} This 1.4 x 1.7-cm CMOS chip incorporates 1.68 million transistors using a 0.75- μ m, three-metal process. Figure 3 shows the chip's block diagram and Figure 4 shows a photograph of the chip itself. The design provides high performance through superscalar (two instruction issue) operation with an exceptionally high frequency internal clock cycle. Production chips and systems are available at clock speeds up to 200 MHz.⁸ Despite the fast internal cycle time, the 21064 provides a flexible external interface that can easily accommodate a range of system designs. These designs are well within the range of standard interface devices due to the on-chip programmable system

clock. System designs can run the CPU at from two to eight times the system clock frequency. Initial system designs range from PC to workstation to supercomputer class. They offer the highest microprocessor-based system performance in the industry as measured by the System Performance Evaluation Corporation (SPEC) suite of benchmark programs. (SPEC benchmarks, a series of programs measuring both speed and throughput, have become the standard for measuring computer performance.)

Cycle time implications. Overall performance involves many factors, but the two controlled primarily by the microprocessor designer are cycle time and the amount of work or instructions completed per cycle. Experience developing an earlier short cycle-time microprocessor¹ combined with simulations of possible design alternatives reinforced the RISC conclusions that the more potent lever was cycle time reduction. Based on the aggressive cycle time goal of typical parts at 150 MHz and fast parts at 200 MHz, the design supports two levels of cache hierarchy. The high speeds require on-chip caches to supply data and instructions at the cycle time rate (5 ns). However, die size and speed constraints limit the maximum size of that cache, which then can reduce performance. A large off-chip, second-level cache can mitigate this effect. The combination provides better overall performance and promises a greater rate of improvement as process density increases. The relative performance gain in increasing a small cache is greater than made available by increasing a large cache. In addition, a

small on-chip cache can scale with CPU cycle time much better than a large off-chip cache, allowing a design to take full advantage of advances in process technology.

To maintain the cycle time goals, we carefully evaluated all potential features—even a slight cycle time slip would likely cost more performance than the feature could give us. This philosophy extended through the ongoing architectural definition to avoid requirements that could limit implementation. For example, we postponed the decision regarding inclusion of the scaled add-and-subtract instructions in the architecture until we could demonstrate that implementations would not incur an adverse internal cycle time hit.

Dual issue. Dual-issue capabilities also exhibit cycle time influences. Rather than allow complete dual-issue flexibility, which only improves performance by an approximate increment of 2 percent over the final design, our design slightly restricts the instruction pairs for multiple issue. Compilers can group dual-issue operations in pairs when possible, but excess code expansion arises due to instruction padding if they are required to always align within the pair as well. When necessary, the hardware swaps pairs capable of dual issue. Hardware also serializes pairs that cannot dual issue to streamline internal control and data paths. All important pairs allowed by combinations of functional units can dual issue.

Since load-and-store operations predominate in RISC codes, the design provides a separate address unit to allow load and stores to execute with operate instructions. Table 1 shows the general instruction pairings for dual issue. There are only two exceptions to these rules. Branches cannot dual issue with stores of the same format because they share a register file port, and stores or branches cannot dual issue with operates of a different format because they share an instruction bus. For example, integer stores cannot dual issue with

Table 1. General dual-issue rules.

Instruction A	Instruction B
Integer operate	Floating-point operate
Load/store	Operate
Branch	Load/store/operate

floating-point operates.

Pipeline. As shown in Figure 5, the integer and floating-point pipelines are, respectively, seven- and 10-stages deep. The first four stages are common to the two pipes and comprise the instruction fetch-and-issue section of the chip. Each stage can process up to two instructions in parallel. In the instruction fetch (IF) stage, the processor fetches a pair of instructions each cycle from the 8-Kbyte instruction cache. The swap (SW) stage controls instruction prefetching, doing branch prediction and cache index calculation as well as the swap or serialization operation described earlier. The issue-zero (I0) stage checks for intrafetch dependencies. This stage also completes the decoding and set up for the issue-one (I1) stage, which includes the register conflict detection and instruction issue to the datapath function units. Both the integer and floating-point register files are read in the I1 stage to supply data to integer, floating-point, load/store, and branch calculation units as shown.

The integer calculation pipeline writes results back to the integer register file in pipe stage 6, while floating-point calculations write to the floating-point register file in stage 9. Pipe stage 4 resolves branches, after which the prefetcher is redirected, resulting in a four-cycle misprediction penalty.

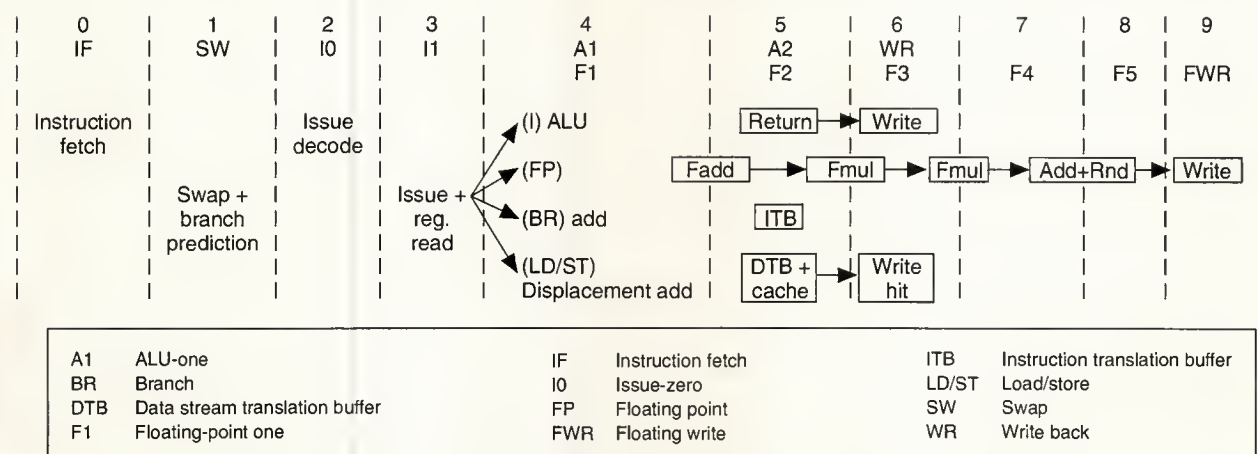


Figure 5. Pipeline.

Branch instructions and condition codes

Branches pose an increasingly severe problem in heavily pipelined and superscalar computer designs as a pipeline flush costs more potential instruction issue slots. The Alpha AXP architecture offers an advantage in handling branch instructions. Most computer architectures include condition codes to hold the result of arithmetic operations that can later be used to determine the outcome of branches. Unfortunately, the condition code register itself can become a point of resource contention if you assume that multiple instructions are executed simultaneously.

Recognizing this problem, some architectures offer combined compare-and-branch instructions that both reduce the number of instructions and eliminate the intermediate condition code storage. These instructions, though, force the arithmetic operation to be performed immediately before the branch. Arithmetic operations typically are one of the last pipeline stages, which therefore increases the misprediction penalty. In a superscalar implementation, it may be possible to resolve the branch decision early and overlap its execution with unrelated instructions.

The Alpha AXP architecture does not use condition codes. Instead, it resolves all branches based on the test of a single register. In effect, any register can hold branch condition information, eliminating the resource problem. In addition, since the branch need only test a single register, all that is required to resolve any branch immediately after reading the register file are the most- and least-significant bits, and a zero detection, which could be stored at register write time.

Loads that hit in the 8-Kbyte on-chip data cache write the associated integer or floating-point register file in pipe stage 6, simultaneously with integer calculations. The chip checks data cache misses in the large off-chip backup cache under complete CPU control. It only generates a system read block command if both caches miss. Instruction cache misses also get checked in the backup cache and fetch a second sequential 32-byte block into an on-chip streaming buffer. If the CPU reports an additional instruction cache miss that hits in the stream buffer, the data is simply moved into the cache and the next block is fetched in parallel with instruction execution.

Despite the apparent two-cycle delay for integer calculations, data is available after the first cycle in most cases. As shown in Figure 6, an extensive set of data bypass paths allows many back-to-back dependent operations to execute

at fully pipelined speeds. In all, the chip uses 45 different bypass paths to minimize the effect of pipeline latency on dependent operations. All register conflict checking is done in hardware. Up to 22 operations thus can be in various stages of completion simultaneously, including 14 within pipeline stages 0 to 6, three in the extended floating-point pipe, three outstanding load misses, a floating-point division, and an integer multiplication.

Branch handling. With such a deep pipeline, branch handling is particularly important. The Alpha AXP architecture reduces branches through the use of the conditional move instructions and also includes hints for hardware-assisted branch prediction. The 21064 uses these hints and includes additional features. The chip can statically predict conditional branches using the sign of the displacement field to predict backward branches as taken and forward branches as not taken. In addition, the 21064 contains a 2K by 1-bit branch history table for dynamic prediction that provides approximately 80 percent accuracy for most programs.

The instruction prefetcher also contains a last-in, first-out stack of recent subroutine return addresses used to predict return paths for subroutines. The stack is repaired during pipeline flushes and therefore allows two additional benefits. As explained in the box, branch misprediction can become a performance issue at these fast cycle times due to the length of the pipeline. The chip uses the subroutine return stack to source the alternate, and correct, branch path one cycle earlier than the program counter datapath could provide it upon branch misprediction. In addition, the stack is used to accurately predict the return address for exceptions, since most exceptions (such as translation look-aside buffer miss) as measured by frequency, return to the original routine.

Integer unit. The integer register file contains thirty-two 64-bit general-purpose registers. It provides six ports, including four reads and two writes to allow the parallel execution of both integer calculations and load, store, or branch operations. The data path includes dedicated adder, shifter, multiplier, and logic units. Both the logic unit and adder provide results in one cycle. The shifter requires two cycles for results but is fully pipelined. The multiplier is not pipelined for area savings, and it supports the Alpha AXP UMULH instruction that returns the upper 64 bits of a 128-bit product for extended-precision operations and integer division support.

Floating-point unit. The floating-point unit combines maximum throughput with short latencies. It contains a 32-entry by 64-bit register file with three read and two write ports. The multiplier uses a radix-8 Booth algorithm in a fully pipelined two-way interleaved array. The rounding operation is completed simultaneously with the last adder stage for all operations. For compatibility, this unit supports both VAX and IEEE single- and double-precision data formats. It can initiate new instructions every cycle with dependent operations requiring six-cycle latency. The fast cycle-time goal trans-

lates into longer total latency as measured in cycles. For many floating-point codes, though, throughput and optimized compiler algorithms deliver exceptional performance as demonstrated by the >200 SPECfp92 values measured on DEC 10000 systems.

Address unit. The address unit performs all load-and-store operations. To do so in parallel with other units, it contains a dedicated displacement adder rather than sharing the integer calculation adder. The address unit contains a 32-entry data translation look-aside buffer. Entries can be used to translate single pages or groups of contiguous pages. The unit allows ranges of 8 Kbytes, 64 Kbytes, 512 Kbytes, or 4 Mbytes for each entry. The address unit can process up to three outstanding load misses to avoid blocking nondependent instructions.

Store instructions aggregate data in a 4-entry \times 32-byte write buffer. The write buffer reduces off-chip bandwidth requirements by merging data from adjacent stores. It also allows early service for critical load data by temporarily delaying stores that would have otherwise occupied the data bus. The AXP architecture allows this reordering to improve performance; the memory barrier instruction can inhibit the reordering when necessary. The address unit allows back-to-back load-and-store operations in any order by accessing the current store tag with the last store data in separate cache tag and data arrays. The address unit supports wrapped reads (target word first) on primary cache misses, while filling 32-byte cache blocks. This minimizes the latency incurred when the return data is immediately needed. If the pipeline was blocked waiting for load data, it can continue as soon as the target word is returned at the same time that it fills the remainder of the cache line in the background.

Pipeline control/exceptions. The pipeline can be interrupted for a number of reasons including branch mispredictions, instruction cache misses, and interruption and exception conditions. Either a conditional branch or calculated jump instruction can produce branch mispredictions. They do not require hardware unrolling because both mispredictions are detected before the write back stage. Interruptions and exceptions cause traps to PALcode. These traps resemble mispredictions but also drain the pipeline before executing the new flow. Hardware reduces the idle pipeline time by overlapping the drain with the prefetch of the new instructions.

Privileged Architecture Library. A unique feature of the AXP architecture is the privileged architecture library. The PAL routines used with the 21064 allow flexibility in the definition of a hardware/software interface by assisting some hardware-related tasks and completely emulating others. For example, the hardware traps to PALcode to parse and service interruptions as well as to update translation look-aside buffers.

A second method of entering PALcode is through explicit CALL_PAL instructions. The chip supports 128 direct hard-

SUBQ	R0, R5, R1	; cycle 0
ADDQ	R1, R2, R3	; cycle 1
CMPLT	R3, #500, R4	; cycle 2
BEQ	R4, Target	; cycle 3

Figure 6. Multiple bypass paths allow many instructions to execute in sequential cycles despite the deep pipeline.

ware dispatches for individual CALL_PAL-type instructions. Upon execution, a CALL_PAL instruction both branches to the selected PAL routine and enables PALmode privileges. These privileges allow PAL routines to access a complete internal state, otherwise hidden from the architected hardware/software interface. PAL can physically access both instruction and data-stream memory by disabling memory mapping, and assure atomic sequences of instructions by disabling interruptions. CALL_PAL routines support a variety of operations, generally too complex to be implemented in hardware. For example, the PAL provides the swap process context operation as a CALL_PAL instruction that can be unique for each operating system.

PALcode routines can be completely customized because they use a superset of the AXP instruction set. The architecture exclusively reserves five operation codes for PAL which allows each implementation to define these instructions for best use. A hardware implementation and PAL routines form a matched set that together make up the operating system and programmer interface. Since only the interface must remain consistent between implementations, future chips have complete flexibility to make low-level hardware trade-offs without impacting existing code. In addition, designers can redefine this interface to meet the needs of each operating system. The 21064 currently supports three operating systems with individual interface requirements.

Performance tuning. In the first production implementation of any new architecture, performance feedback is important for both software tuning and future hardware projects. With improving integration and cycle times, this information is increasingly difficult to obtain at the pin interface, or is so far removed from program execution that it is of little value. The 21064 contains two methods of providing more relevant information directly from running systems.

First, the Alpha AXP architecture offers a cycle counter capable of recording absolute and process virtual times at very fine intervals (single cycle on 21064). Its use, however, requires code modification. Second, the 21064 contains on-chip performance counters that count selected events and produce interruptions upon counter overflow. Through the use of PALcode and operating system utilities, these counters can collect data on unmodified applications. The design provides two counters that can select from a variety of sources

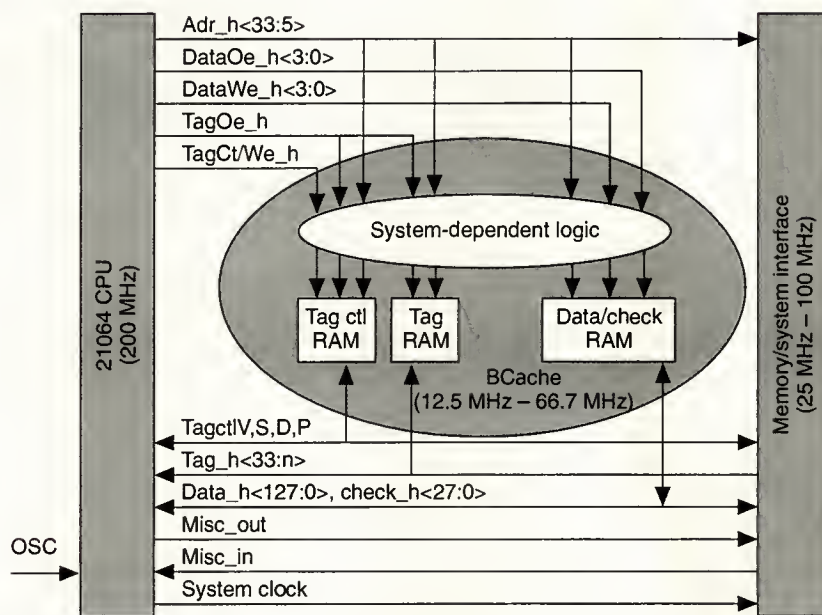


Figure 7. Chip interface. Three time domains exist in a typical system, with the CPU executing out of the internal caches at up to 200 MHz, the backup cache loop ranging from one third to one sixteenth of the CPU frequency, and the remaining system logic executing at one half to one eighth of the CPU frequency.

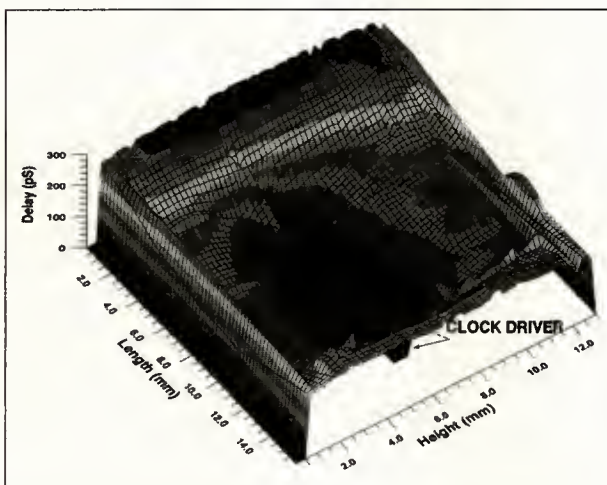


Figure 8. Alpha clock delay.

including instruction issue, pipeline stalls, and instruction mix as well as cache miss, branch misprediction, and two input pins that can be further broken down to gather external system data.

Interface. To accommodate a range of system designs, the interface is extremely flexible. See Figure 7 for a drawing of the chip interface. Although the chip operates with a 3.3-volt power supply, it can also interface with more common 5-volt logic. Data bus widths of 128 and 64 bits for reduced-cost systems are available, and the system clock speed can be set at any submultiple of the CPU speed from one half to one eighth. We have designed a 25-MHz EISA bus-based system that uses a one-to-six CPU clock divisor using standard PC interface parts. Even at 150-MHz CPU operation, cooling only requires a heat sink.

The chip supports up to 16 Gbytes of physical memory and an optional second-level back-up cache ranging in size from 128 Kbytes up to 16 Mbytes. The cache access path is combinatorial. It can support a variety of static RAM speeds, selectable through an on-chip register at 3 to 16 times the CPU clock cycle time. The interface supports parity or ECC protection. In the event of a back-up cache miss, the chip issues a

system command to perform the necessary read or write operation. System commands interact with board logic in a handshake manner and operate at the selected system clock multiple, not the CPU clock speed.

Multiprocessing. The chip provides multiprocessing support in a flexible manner. Through valid, dirty, and shared (write protected) tag control signals, we can configure a write-back external cache on the 21064 to support a variety of cache coherence policies. Digital's systems use a conditional write-through policy although the chip can also support an ownership policy. Internal cache invalidate controls allow a system to maintain coherence of the internal cache. Through pin support for maintaining a backmap, the system can also implement cache invalidate filtering based on the contents of the primary cache if desired.

Clocking. Since it operates at speeds of up to 200 MHz, designing the 21064 required us to rethink many aspects of CMOS circuit design. Most critical was the decision to use a single-wire, two-phase clocking scheme. This type of clocking helps to eliminate dead time between phases. To ensure correct latching operation, though, the clock edge rate had to be extremely fast to avoid race-through of the latch data. Our solution included a very large clock driver with a final stage containing 10-11/64-inch-wide PMOS and 4-5/64-inch-wide NMOS devices. The driver switches the clock load in

Table 2. Measured performance under OpenVMS AXP V1.

	DEC 3000 Model 400	DEC 3000 Model 500	DEC 4000 Model 610	DEC 7000 Model 610	DEC 10000 Model 610
CPU frequency (MHz)	133	150	160	182	200
BCache size (Kbytes)	512	512	1,000	4,000	4,000
TPC-A (Rdb v.6)*	NA	NA	NA	302	NA
SPECint92	63.8	72.6	81.2	94.8	104.3
SPECfp92	112.2	126.0	143.1	182.1	200.4
SPECrate_int92	NA	NA	NA	NA	NA
SPECrate_fp92	2,631.6	2,967.4	3,317.1	4,126.0	
2 process			6,214.5	8,135.1	
3 process				11,859.8	
4 process				15,739.4	17,187.2
Linpack 100 × 100**	26.4	30.2	36.3	38.6	42.5
1,000 × 1,000**	90	107	114	141	155
Perfect BM suite†	18.1	20.4	22.9	26.0	28.6
Cernlib (CERN units)	16.9	19.0	21.0	23.6	26.0
Livermore loops	18.7	21.3	22.9	25.6	28.1
Slalom patches	5,644	6,022	6,384	7,018	7,248
SPECint89	65.8	73.5	83.7	95.1	104.5
SPECfp89	150.6	169.9	188.4	244.2	268.6
SPECmark89	108.1	121.5	136.2	167.4	184.1

* Transactions per second
 ** Project linear scaling for microprocessor configurations
 † Geometric mean

0.5 ns, drawing a peak switching current of 43A. We extensively analyzed the clock both to ensure the integrity of the supply voltage during switching and to guarantee that the adjacent latches saw very little clock skew for proper operation. To address the supply voltage problem, we added 0.13 μ F of on-chip decoupling capacitance. This was sufficient to supply all the charge associated with a complete CPU cycle with only 10-percent degradation of the supply voltage.

The skew problem required analysis of the 1.2-million-element RC clock grid. We used a simulator derived from the Carnegie-Mellon AWESim circuit simulation program to examine the grid at 10-ps intervals. As shown in Figure 8, a monotonic clock wave propagates outward from the center clock driver. Any inward movement of the wave or large discrepancies would indicate potential timing hazards in the design. Such analysis proved necessary for correct operation of the chip, as early simulation results did, in fact, identify errors in the grid connection.

System performance

Performance tuning is an ongoing effort with work continuing in both compiler algorithms and optimizations as well

as system tuning. However, as shown in Tables 2 and 3 (next page), initial data demonstrate excellent performance. These tables include results over a variety of commonly used benchmarks under both OpenVMS AXP V1 and DEC OSF/1 V1.2. Both SPECint92 and SPECfp92 values establish a new high point for system performance. Linpack among other floating-point intensive benchmarks demonstrates impressive floating-point capability, with the 1,000 × 1,000 values representing greater than 75 percent of the peak theoretical rate. Integer performance is equally impressive—the DEC 3000 Model 500X workstation achieves a SPECint92 rating of over 110. The more recent SPECrate benchmarks show nearly linear scaling across all multiprocessor configurations as demonstrated by the SPECrate data run under OpenVMS. [For a fuller description of SPEC benchmarks, see H.G. Sachs et al., "Design and Implementation Trade-offs in the Clipper 400 Architecture," *IEEE Micro*, Vol. 11, No. 3, June 1991, pp. 18-21, 74-80.—Ed.]

Table 4 shows results of benchmark performance for translated VAX images. The goal of the translation effort was to match or exceed the performance of similarly priced VAX systems. We met this goal, and many VAX user applications

Table 3. Measured performance under DEC OSF/1 V1.2.

	DEC 3000 Model 400	DEC 3000 Model 500	DEC 3000 Model 500X	DEC 4000 Model 610	DEC 7000 Model 610	DEC 10000 Model 610
CPU frequency (MHz)	133	150	200	160	182	200
BCache size (Kbytes)	512	512	512	1,000	4,000	4,000
SPECint92	74.7	84.4	110.9	94.6	103.1	116.5
SPECfp92	112.5	127.7	164.1	137.6	176.0	193.6
SPECrate-int92	1,763.0	1,997.0	2,611.0	2,198.0	2,571.7	2,765.0
SPECrate-fp92	2,662.0	3,023.0	3,910.0	3,247.0	4,178.7	4,368.4
Linpack 100 × 100	26.0	29.6	39.8	35.0	36.9	40.5
1,000 × 1,000*	91.7	103.5	133.2	110.1	137.8	151.1
X11perf (2D Kvec/s)	579.0	662	670	NA	NA	NA
X11perf (2D Mpix/s)	27.2	31.0	31.0	NA	NA	NA
Dhrystones/s V1.1	235,939	266,487	349,785	297,345	330,577	363,743
V2.1	238,095	263,157	333,333	294,117	333,333	357,142
Perfect BM suite**	18.4	20.7	26.2	23.1	26.4	29.2
Cernlib (CERN units)	18.8	21.3	28.9	23.2	26.0	29.0
Livermore loops**	17.4	19.5	26.3	22.3	25.4	27.8
Slalom patches	5,776	6,084	7,134	6,496	6,902	7,248
SPECint89	73.1	83.1	108.6	92.9	107.4	116.2
SPECfp89	141.7	162.8	208.9	177.0	249.8	275.8
SPECmark89	111.1	126.1	160.8	137.3	175.5	192.1

*64 bit, double precision
**Geometric mean

Table 4. Measured performance of VAX translated code under OpenVMS.

	DEC 7000 Model 610 Alpha AXP*	VAX 7000/610	DEC 3000 Model 500 Alpha AXP*	VAX 4000/90
SPECmark-89	44.43	42.09	34.37	32.77
SPECint-89	26.71	31.48	20.74	26.71
SPECfp-89	62.36	51.08	48.14	37.55

*Translated with DECmigrate

report between 10 to 15 percent faster times running the translated image on the AXP platform.

Table 5 shows results for translated MIPS images using DECmigrate. As can be seen, performance of the translated image approaches that of native compiled code on the DEC 3000 Model 500 system running DEC OSF/1.

THE ALPHA AXP ARCHITECTURE marks a new beginning for Digital. The combination of binary translation and PALcode affords the luxury of starting fresh, while maintaining strong compatibility with existing code. The architecture provides for growth in multiple fields as well as flexibility in the implementation of exception handling and operating system specific state. We carefully considered trends in computing such as multiple-instruction issue and multiprocessing to avoid restrictive requirements on future systems. Finally, 64-bit data capability and a 64-bit linear address space, offering over four billion

times the range of a 32-bit space, should provide ample power and programming flexibility for years to come.

The goal of the chip design was to deliver the highest performance single-chip microprocessor in the industry, capable of forming the core of a range of systems from PC class to high-end server. Benchmark results attest to that accomplish-

ment. A wide range of system designs are currently available and expanding. In fact, the 21064 chip also forms the basis for the announced massively-parallel processing (MPP) supercomputer from Cray Research, Inc. With the availability of Microsoft Windows NT and native Novell NetWare, the architecture will offer an easy bridge for adding PC applications to the growing list of over 2,500 OpenVMS and Unix applications available today.

Unlike our previous architectures, Alpha AXP is an open computer architecture. Mitsubishi Electric Corp. recently joined over 35 other corporate Alpha AXP partners at all levels of design integration and will offer a second source for the 21064 as well as new designs in the future. Within Digital, we are currently designing multiple chips. High-performance parts include a speed enhanced version of the 21064 that will double the internal cache sizes and a next-generation, quad-issue processor. Design of a high integration device for reduced system cost is also in progress.

With the demonstrated performance of the current designs, availability of software, and a growing list of suppliers, the Alpha AXP architecture is well positioned for the future. ■

Acknowledgments

Many people contributed to the design of the Alpha AXP architecture, with a large majority of the current definition attributable to principal architects Dick Sites and Rich Witek. The 21064 design and verification team consisted of Dan Dobberpuhl, Rich Witek, Randy Allmon, Rob Anglin, Dave Bertucci, Sharon Britton, Linda Chao, Rob Conrad, Dan Dever, Bruce Gieseke, Soha Hassoun, Greg Hoepfner, John Kowaleski, Kathy Kuchler, Maureen Ladd, Mike Leary, Liam Madden, Ed McLellan, Dirk Meyer, Jim Montanaro, Don Priore, Vidya Rajagopalan, Sri Samudrala, Sri Santhanam, Scott Kreider, Stephan Meier, Andy Payne, Homayoon Akhiani, Mike Kantrowitz, and Dave Conroy, as well as others, who devoted tremendous amounts of their time and talents to ensure the successful completion of the design. Jim Montanaro, Mark Cooley, John Faricelli, and John Kowaleski created the clock analysis and graphical output.

References

1. R.L. Sites, "Alpha AXP Architecture," *Comm. ACM*, Vol. 36, No. 2, Feb. 1993, pp. 33-44.
2. *Alpha Architecture Reference Manual*, R.L. Sites, ed., Digital Press, Bedford, Mass., 1992.
3. R. Conrad et al., "A 50-MIPS (Peak) 32/64-Bit Microprocessor," *Digest Tech. Papers, IEEE Solid-State Circuits Conf.*, Piscataway, N.J., Feb. 1989, pp. 76-77.

Table 5. MIPS translated code performance on DEC 3000 Model 500 (DEC OSF/1).

	Native	DECmigrate
SPECint92	84.4	68.6 (81%)
SPECfp92	127.7	81.9 (64%)

4. *VAX Architecture Reference Manual, Second Ed.*, R. Brunner, ed., Digital Press, Bedford, Mass., 1991.
5. *Cray-1 Computer System Reference Manual*, Form 2240004, Cray Research, Inc., 1977.
6. D. Dobberpuhl et al., "A 200-MHz 64-Bit Dual-Issue CMOS Microprocessor," *IEEE J. Solid-State Circuits*, Vol. 27, No. 11, Nov. 1992, pp. 1555-1567.
7. D. Dobberpuhl et al., "A 200-MHz 64-Bit Dual-Issue CMOS Microprocessor," *Digest Tech. Papers, IEEE Solid-State Circuits Conf.*, Feb. 1992, pp. 106-107.
8. *DECchip 21064-AA Microprocessor Hardware Reference Manual*, Digital Press, Bedford, Mass., October 1992.
9. R.L. Sites et al., "Binary Translation," *Comm. ACM*, Vol. 36, No. 2, Feb. 1993, pp. 69-81.



Edward McLellan is a principal engineer in Digital's Semiconductor Engineering Group. He has contributed to multiple chip designs, including PDP-11 and VAX floating-point processors, prior to the Alpha project. He was a coarchitect of the 21064 processor.

McLellan received a BS in computer and systems engineering from Rensselaer Polytechnic Institute and has done graduate work at the University of Massachusetts. Currently, he is working on a third-generation Alpha AXP processor.

Direct any questions concerning this article to the author at the Digital Equipment Corporation, HLO2-3/J03, Hudson, MA 01749; mclellan@ad.enet.dec.com.

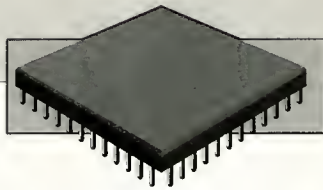
Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 156

Medium 157

High 158



Sparcle: An Evolutionary Processor Design for Large-Scale Multiprocessors

Working jointly at MIT, LSI Logic, and Sun Microsystems, designers created the Sparcle processing chip by evolving an existing RISC architecture toward a processor suited for large-scale multiprocessors. This chip supports three multiprocessor mechanisms: fast context switching, fast, user-level message handling, and fine-grain synchronization. The Sparcle effort demonstrates that RISC architectures coupled with a communications and memory management unit do not require major architectural changes to support multiprocessing efficiently.

Anant Agarwal

John Kubiawicz

David Kranz

Beng-Hong Lim

Donald Yeung

Massachusetts Institute of
Technology

Godfrey D'Souza

LSI Logic

Mike Parkin

Sun Microsystems

The Sparcle chip clocks at no more than 40 MHz, has no more than 200,000 transistors, does not use the latest technologies, and dissipates a paltry 2 watts. It has no on-chip cache, no fancy pads, and only 207 pins. It does not even support multiple-instruction issue. Then why do we think this chip is interesting?

Sparcle is a processor chip designed to support large-scale multiprocessing. We designed its mechanisms and interfaces to provide fast message handling, latency tolerance, and fine-grain synchronization. Specifically, Sparcle implements

- *Mechanisms to tolerate memory and communication latencies, as well as synchronization latencies.* Long latencies are inevitable in large-scale multiprocessors, but current microprocessor designs are ill-suited to handle such latencies.
- *Mechanisms to support fine-grain synchronization.* Modern microprocessors pay scant attention to this aspect of multiprocessing, usually providing just a test-and-set instruction, and in some cases, not even that.
- *Mechanisms to initiate communication actions to remote processors across the communications network, and to respond rapidly to asynchronous events such as synchronization*

faults and message arrivals. Current microprocessor designs do not support a clean communications interface between the processor and the communications network. Furthermore, traps and other asynchronous event-handlers are inefficient on many current microprocessors, often requiring tens of cycles to reach the appropriate trap service routine.

The impetus for the Sparcle chip project was our belief that we could implement a processor that provides interfaces for the above mechanisms by making small modifications to an existing microprocessor. Indeed, we derived Sparcle from Sparc¹ (scalable programmable architecture from Sun Microsystems), and we integrated it into Alewife,^{2,3} a large-scale multiprocessor system being developed at MIT.

Sparcle tolerates long communication and synchronization latencies by rapidly switching to other threads of computation. The current implementation of Sparcle can switch to another thread of computation in 14 cycles. Slightly more aggressive modifications could reduce this number to four cycles. Sparcle switches to another thread when a cache miss that requires service over the communications network occurs, or when a synchronization fault occurs. Such a processor requires a pipelined memory and communications

system. In our system, a separate communications and memory management chip (CMMU) interfaces to Sparcle to provide the desired pipelined system interface. Our system also provides a software prefetch instruction. For a description of the modifications to a modern RISC microprocessor needed to achieve fast context switching, see our discussion under architecture and implementation of Sparcle later in the article.

Sparcle supports fine-grain data-level synchronization through the use of full/empty bits, as in the HEP computer.⁴ With full/empty bits, a lock and access of the data word protected by the lock can be probed in one operation. If the synchronization attempt fails, the synchronization trap invokes a fault handler. In our system, the external communications chip detects synchronization faults and alerts Sparcle by raising a trap line. The system then handles the fault in software trap code.

Finally, Sparcle supports a highly streamlined network interface with the ability to launch and receive interconnection network messages. While this design implements the communications interface with the interconnection network in a separate chip, the CMMU, future implementations can integrate this functionality into the processor chip. Sparcle supports rapid response to asynchronous events by streamlining Sparc's trap interface and by supporting rapid dispatch to the appropriate trap handler. To achieve this, Sparcle provides two special trap lines for the most common types of events—cache misses to remote nodes and synchronization faults. Sparcle uses a third trap line for all other types of events. Also, this chip has an increased number of instructions in each trap dispatch entry so that vital trap codes can be put in line at the dispatch points.

Sparcle's design process was unusual in that it did not involve developing a completely new architecture. Rather, we implemented Sparcle with the help of LSI Logic and Sun Microsystems by slightly modifying the existing Sparc architecture. At MIT, we received working Sparcle chips from LSI Logic on March 11, 1992. These chips have already undergone complete functional testing. We are currently continuing to implement the Alewife multiprocessor so that we can thoroughly evaluate our ideas and subject the Sparcle chips to full-speed testing. Figure 1 shows an Alewife node with the Sparcle chip.

Mechanisms for multiprocessors

By supporting the widely used shared-memory and message-passing programming models, Sparcle eases the programmer's job and enhances parallel program performance. We have implemented programming constructs in parallel versions of Lisp and C that use these features. Sparcle's features fall into three areas, the first two of which support the shared-memory model:

- **Fine-grain computation.** Efficient support of fine-grain expression of parallelism and synchronization can en-

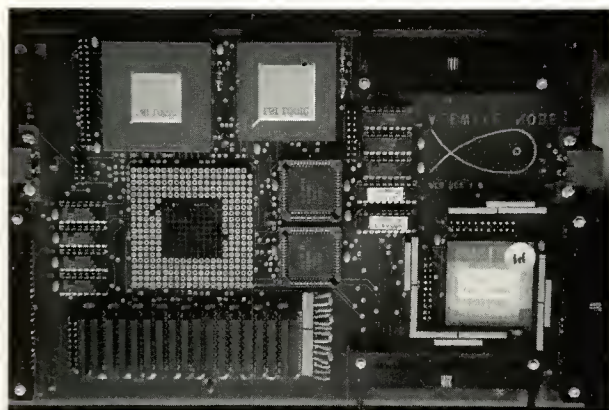


Figure 1. An Alewife node.

hance performance by increasing parallelism and reducing communication overhead. This enhancement relieves the programmer of undue effort in partitioning data and controlling flow into coarser chunks to increase performance.

- **Memory latency tolerance.** Context switching and data prefetching can reduce communication overhead introduced by network delays. For shared-memory programs, the switch must be very fast and occur automatically when a remote cache miss occurs.
- **Efficient message interface.** The ability to send and receive messages is needed to support message-passing programs. Such interfacing can also improve the performance of shared-memory programs in some common situations.

Before we can examine the implementation of these features in Sparcle, we need to consider each of these areas in turn, and discuss why they are useful for large-scale multiprocessing.

Fine-grain computation. As multiprocessors become larger, the grain size of parallel computations decreases to satisfy higher parallelism requirements. Computational grain size refers to the amount of computation between synchronization operations. Given a fixed problem size, the overhead of parallel and synchronization operations limits the ability to use a larger number of processors to speed up a program. Systems supporting fine-grain parallelism and synchronization attempt to minimize this overhead so that parallel programs can achieve better performance.

The challenge of supporting fine-grain computation is in implementing efficient parallelism and synchronization constructs without incurring extensive hardware cost, and without reducing coarse-grain performance. By taking an evolutionary approach in designing Sparcle, we have attempted to satisfy these requirements.

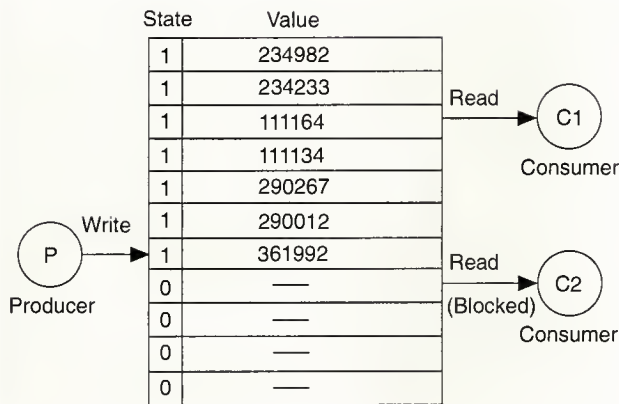


Figure 2. J-structures.

We can express fine-grain parallelism and synchronization at the data level (data-level parallelism) or at the thread level (control-level parallelism).

Data-level parallelism. Data-level parallelism and synchronization allows the program to synchronize at the level of the smallest possible unit—a memory word. At the programming language level, we provide parallel do-loops to express data-level parallelism, and J-structure and L-structure arrays to express fine-grain data-level synchronization.

Inspired by the I-structures of Arvind, Nikhil, and Pingali,⁵ the J-structure is a data structure for producer-consumer style synchronization. It is like an array, but each element has an additional state—full or empty. The initial state of a J-structure element is empty. A reader of an element waits until the element's state is full before returning the value. A writer of an element writes a value, sets the state to full, and signals waiting readers to proceed. A write to a full element signals an error. For efficient memory allocation and cache performance, J-structure elements can be reset to an empty state. Figure 2 illustrates how J-structures can be used for data-level synchronization.

In the example of Figure 2, producer P is sequentially filling in the elements of a J-structure. Consumer C1 reads an element that is already filled and immediately gets its value. Consumer C2 reads an empty element and thus has to wait for P to write the element. Since we are synchronizing at the level of individual elements, both C1 and C2 can access the elements of the J-structure without waiting for P to completely fill all the elements of the J-structure.

L-structures are similar to J-structures but support three operations: a locking read, a nonlocking read, and a synchronizing write. A locking read waits until the element is full before emptying it (that is, locking it) and returning the value. A nonlocking read operation also waits until the element is full, but returns the value without emptying the ele-

ment. A synchronizing write stores a value to an empty element and sets it to full, releasing any waiters. An L-structure thus allows mutually exclusive access to each of its elements and allows multiple nonlocking readers.

Sparcle supports J- and L-structures, as well as other types of fine-grain data-level synchronization, with per-word, full/empty bits in memory.⁴ Sparcle provides new load/store instructions that interact with the full/empty bits. The design also includes an extra synchronous trap line to deliver the full/empty trap. This extra line allows Sparcle to immediately identify the trap.

Control-level parallelism. Control-level parallelism may be expressed by wrapping *future* around an expression or statement *X*. The *future* keyword declares that *X* and the continuation of the future expression may be evaluated concurrently. Fine-grain support allows the amount of computation needed for evaluating *X* to be small without severely affecting performance.

If the compiler or runtime system chooses to create a new task to evaluate *X*, it also creates an object known as a *placeholder* that is returned as the value of the future expression. The placeholder is created in an undetermined state. Evaluation of *X* yields its value and determines the placeholder. Any task that attempts to use the value of *X* before *X* has been completely evaluated will encounter the undetermined placeholder and will suspend operation until the placeholder is determined.

This functionality is implemented using (by software convention) the low bit of a data value as a placeholder tag; that is, a pointer to a placeholder has the low bit set and all other values have the low bit clear. New add, subtract, and compare instructions in Sparcle trap if the low bit of any operand is set. Likewise, dereferencing a pointer with the low bit set will cause an address alignment trap to a similar routine. If the trap handler can determine the value at the placeholders, it places this value in the target register, and normal execution resumes. Otherwise, the trapping task waits until the value of the placeholder becomes available.

With this support, a compiler can generate code without knowing which data values may be computed concurrently. Consequently, Sparcle incurs no runtime overhead to ensure the detection of placeholders.

Memory latency tolerance. Since memory in large-scale multiprocessors is distributed, cache misses to remote locations will incur long latencies and potentially reduce processor use. Figure 3 illustrates this problem by depicting processor and network activity when a single thread executes on the processor. When the thread suffers a long-latency cache miss, the processor waits for the miss to be satisfied before it can proceed. While waiting, both the processor and the network suffer idle time, thereby reducing their effective usage. Using latency tolerance mechanisms alleviates this problem and helps improve processor and network usage.

The general class of latency tolerance solutions all implement mechanisms that allow multiple outstanding memory transactions and can be viewed as a way of pipelining the processor and the network. The key difference between this pipeline into the network and the processor's execution pipeline is that the latency associated with the communication pipeline cannot be predicted easily at compile time. A compiler then has difficulty scheduling operations for maximal resource use. Systems must implement dynamic pipelines into the network in which the hardware ensures that multiple, previously issued memory operations have completed before issuing operations that depend on their completion. Context switching is one mechanism for dynamic pipelining. Other methods include prefetching and weak ordering.⁶⁻⁸

Sparcle implements fast context switching as its primary mechanism for dynamic latency tolerance. (Sparcle and its memory controller provide nonbinding prefetch instructions as well.) As illustrated in Figure 4, the basic idea is to overlap the latency of a memory request from a given thread of computation with the execution of a different thread. In the figure, when thread 1 suffers a cache miss, the processor switches to thread 2, thereby overlapping the cache miss latency of thread 1 with useful computation from thread 2.

In Alewife, when a thread issues a remote transaction or suffers an unsuccessful synchronization attempt, the Alewife CMMU traps the processor. If the trap resulted from a cache miss to a remote node, the trap handler forces a context switch to a different thread. Otherwise, if the trap resulted from a synchronization fault, the trap handling routine can switch to a different thread of computation. For synchronization faults, the trap handler might also choose to retry the request immediately (spin).

Processors that switch rapidly between multiple threads of computation are called multithreaded architectures. The prototypical multithreaded machine is the HEP. In the HEP, the processor switches every cycle between eight processor-resident threads. Cycle-by-cycle interleaving of threads is termed fine multithreading. Although fine multithreading offers the potential for high processor usage, it results in relatively poor single-thread performance and low processor use when there is not enough parallelism to fill all the hardware contexts.

In contrast, Sparcle employs block multithreading or coarse multithreading. That is, context switches occur only when a thread executes a memory request that must be serviced by a remote node in the multiprocessor, or on a failed synchronization request. Thus, a given thread continues to execute as long as its memory requests hit in the cache or can be serviced by a local memory mod-

ule, and as long as synchronization attempts are successful. Block multithreading thus allows a single thread to benefit from the maximum performance of the processor. For multithreading to be useful in tolerating latency, however, the time required to switch to another thread must be shorter than the time to service a remote request. This requires multiple register sets or some other hardware-supported mechanism.

Efficient message interface. An efficient message interface that allows the processor to access the interconnection network directly makes some parallel operations significantly more efficient than if they were implemented solely with shared-memory operations. Examples include remote thread creation and barrier synchronization. With a fast message in Alewife, we can create a thread on a remote processor in 7 μ s. Restricting ourselves to shared-memory operations, remote thread creation takes 24 μ s. Kranz and associates⁹ have studied the importance of an efficient message interface in a shared-memory setting.

In Sparcle, we accomplish a fast message send operation by using the cache bus and coprocessor interface to store data in registers directly into the network, and to load data from the network directly into registers. Two new load/store instructions handle the loading and storing. Sparcle also supports direct memory access for larger messages.

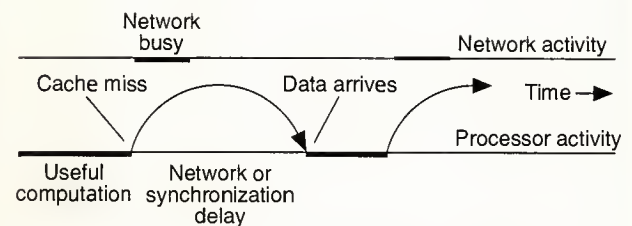


Figure 3. Processor and network activity when a single thread executes on the processor and no latency tolerance mechanisms are employed.

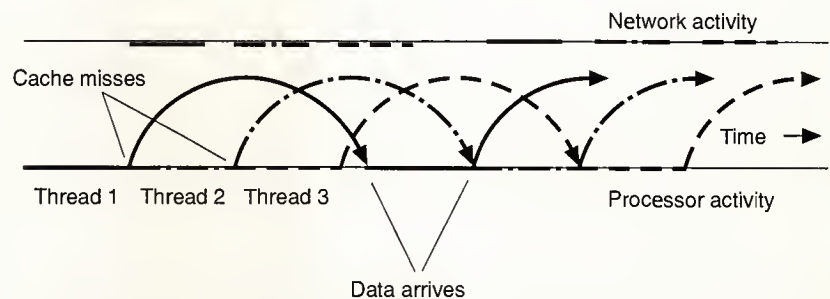


Figure 4. Processor and network activity when multiple threads execute on the processor and fast context switching is used for latency tolerance.

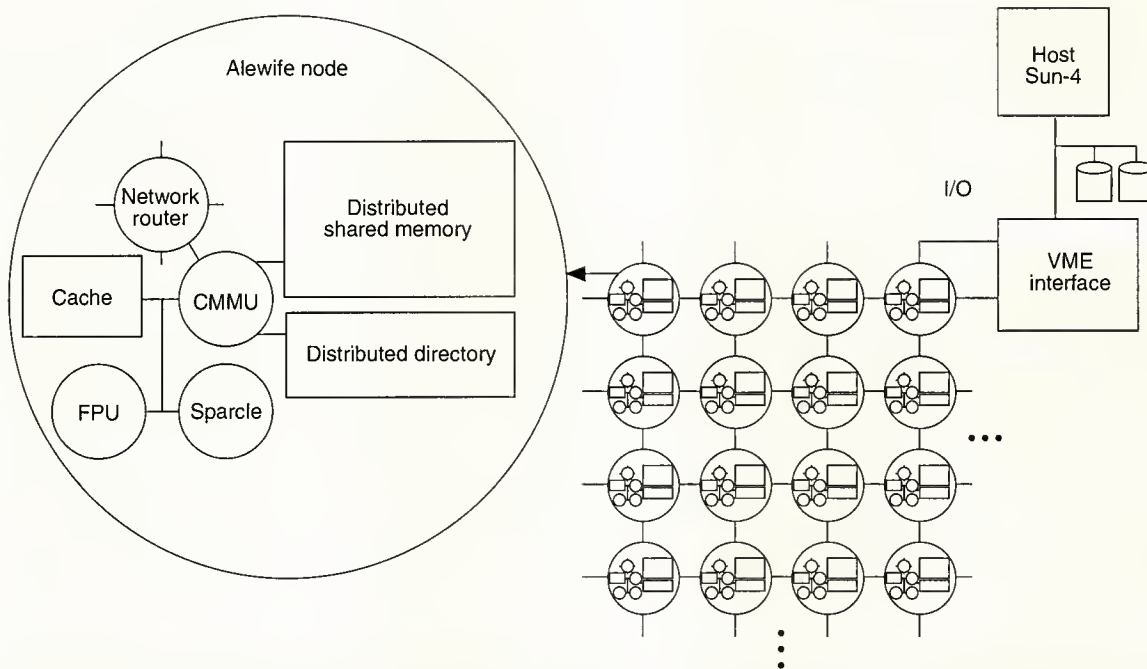


Figure 5. Structure of the Alewife machine.

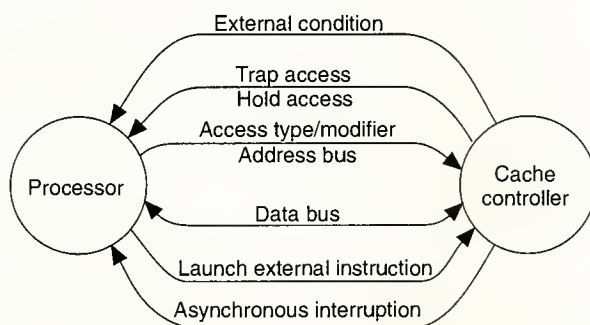


Figure 6. Interface between the processor pipeline and memory controller.

Alewife machine interfaces

The Sparcle chip is part of a complete multiprocessing system. It serves as the CPU for the Alewife machine²—a distributed shared-memory multiprocessor with up to 512 nodes and hardware-supported cache coherence. Figure 5 depicts the Alewife machine as a set of processing nodes connected in a mesh topology. Each Alewife node consists of a processor, a 64-Kbyte cache, a 4-Mbyte portion of globally-shared distributed memory, a CMMU, a floating-point coprocessor, and a network switch. An additional 4 Mbytes of local memory holds

the coherence directory, code, and local data. The network switch chip is an Elko-series mesh routing chip (EMRC) from Caltech that has 8-bit channels. The network operates asynchronously with a switching delay of 30 ns per hop and 60 Mbytes/s through bidirectional channels.

The single-chip CMMU performs a number of tasks, including cache management, DRAM refresh and control, message queuing, remote memory access, and direct memory access. It also supports the LimitLESS cache-coherence protocol,¹⁰ which maintains a few pointers per memory block in hardware (up to five in Alewife) and emulates additional pointers in software when needed. Through this protocol, all the caches in the system maintain a coherent view of global memory.

Sparcle implements a powerful and flexible interface to the CMMU. As depicted in Figure 6, this interface couples the processor pipeline with the CMMU. The interface can be divided into two general classes of signals: flexible data access mechanisms and flexible instruction extension mechanisms.

Together, the Access Type, Address Bus, Data Bus, and Hold Access line form the nucleus of data access mechanisms and comprise a standard external cache interface. To permit the construction of other types of data accesses for synchronization, we have supplemented this basic interface with three classes of signals:

- A *Modifier* that is part of the operation code for load/

store instructions and that is *not* interpreted by the core processor pipeline. The modifier provides several "flavors" of load/store instructions.

- Two *External Conditions* that return information about the last access. They can affect the flow of control through special branch instructions.
- Several vectored memory exception signals (denoted *Trap Access* in the figure). These synchronous trap lines can abort active load/store operations and can invoke function-specific trap handlers.

These mechanisms permit us to extend the load/store architecture of a simple RISC pipeline with a powerful set of operations.

An instruction extension mechanism permits us to augment the basic instruction set with external functional units. Instructions that are added in this way can be pipelined in the same fashion as standard instructions. To make this work, Sparcle reserves a special range of opcodes for external instructions. Also, the memory controller fetches new instructions from the cache bus at the same time that the processor does. Consequently, when the processor decodes an instruction in this range, it asserts the Launch External Inst signal, telling the CMMU to begin execution of the last fetched instruction. Note that the coprocessor interfaces of several microprocessors already provide this functionality.

We contend that we can design such a powerful interface between the processor pipeline and the communications and memory management hardware without significantly modifying the core RISC pipeline of contemporary processors. With this interface in mind, we first discuss several efficient multiprocessor mechanisms that are provided by the Sparcle processor. Later we touch upon the support which the memory controller must provide for these mechanisms.

Sparcle architecture and implementation

Sparcle is best described as a conventional RISC microprocessor with a few additional features to support multiprocessing. These features include support for latency tolerance, support for fine-grain synchronization, and support for fast message handling. Before we describe how we implemented them in the Sparcle processor, we need to discuss these features. Then we can indicate how they can also be implemented in other RISC microprocessors.

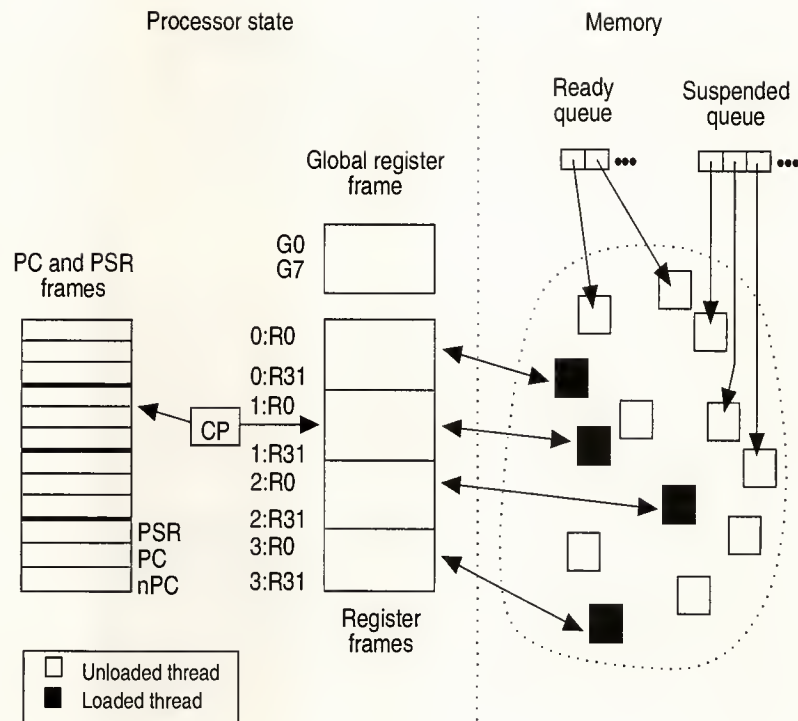


Figure 7. Block multithreading and virtual threads.

Mechanisms for latency tolerance. Figure 7 illustrates fast context switching on a generic processor. This diagram shows four separate register sets with associated program counters and status registers. Each register set represents a context. A hardware register called the context pointer or CP points to the active context. Consequently, a hardware context switch requires only that the context pointer be altered to point to another context. (Depending on details of the implementation, some number of cycles may be needed to flush the pipeline before executing a new context.) This figure also shows four threads actively loaded in the processor. These four threads are part of a much larger set of runnable and suspended threads that the runtime system maintains.

Implementation of fast context switching in Sparcle. In a similar fashion, Sparcle uses multiple register sets to implement fast context switching. The particular Sparcle design that we modified has eight overlapping register windows. Rather than using the register windows as a register stack, we used them in pairs to represent four independent, nonoverlapping contexts. We use one as a context for trap and message handlers, as described by Dally et al.¹¹ and Seitz et al.,¹² and the other three for user threads. The Sparcle current window pointer (CWP) serves as the context pointer. Further, the window

```

RDPSR R16 ; Save PSR in reserved register.
NEXTF R0, R0, R0 ; Move to next active context.
WRPSR R16 ; Restore PSR from other context.
JMLP R17, R0 ; Restore PC
RETT R18, R0 ; Restore nPC and return from trap.

```

Figure 8. Context switch trap code for Sparcle

Cycle	Operation
0	Fetch of data instruction (load or store)
1	Decode of data instruction (load or store)
2	Execute instruction (compute address)
3	Data cycle (which will fail)
→4	Pipeline freeze, indicate exception to processor
5	Pipeline flush (save PC)
6	Pipeline flush (save nPC, decrease CWP)
7	Fetch: RDPSR PSRREG (save PSR in reserved register)
8	Fetch: NEXTF (advance CWP to next active context using WIM)
9	Fetch: WRPSR PSRREG (restore PSR for new context)
10	Fetch: JMLP R17 (load PC, return from trap and)
11	Fetch: RETT R18 (reexecute trapping instruction)
12	Dead cycle from JMLP
13	First fetch of new instruction
14	Dead cycle from RETT (folded into switch time)

Figure 9. Anatomy of a context switch in Sparcle.

invalid mask (WIM) indicates which contexts are disabled and which are active. This particular use of register windows does not involve any modifications, just a change in software conventions.

Unfortunately, the Sparc processor does not have four sets of program counters and status registers. Since adding such facilities would impact the pipeline significantly, we implemented rapid context switching via a special trap with an extremely short trap handler. Thus, when the processor attempts to access a remote memory location that is not in the local cache, the CMMU causes a synchronous memory fault to Sparcle, while simultaneously sending a request for data to the remote node. The trap handler then saves the old program counter and status register, switches to a new context, restores a new program counter and status register, returns from the trap to begin execution in the new context.

With the goal of shortening this trap handler as much as possible, we made the following modifications to the Sparc architecture:

- So that the processor traps immediately to the context-switch code without having to decode the trap type, we added an extra synchronous trap line (with corresponding trap vector).
- We added a new instruction called NEXTF. It is much

like the Sparc SAVE instruction except that the window pointer is advanced to the next active context as indicated by the window invalid mask register. If no additional contexts are active, it leaves the window pointer unchanged.

- We increased the number of instructions for each entry in the Sparc trap vector from 4 to 16. This allows the context switch and other small trap handlers to execute in the trap vector directly.
- We made the value of the current window pointer available on external pins. Among other things, this permits the emulation of multiple hardware contexts in the Sparc floating-point unit by modifying floating-point instructions in a context-dependent fashion as they are loaded into the FPU and by maintaining four different sets of condition bits. Consequently, the context-switch trap handler does not have to worry about the FPU.

Figure 8 shows the context-switch trap handler with these changes. When the trap occurs, Sparcle switches *one* window backward (as does a normal Sparc). This switch places the window pointer *between* active contexts, where the Alewife runtime system reserves a few registers for the context state. As with normal Sparc trapping behavior, the hardware writes the PC and nPC to registers R17 and R18. This trap code places the processor status register (PSR) in register R16.

As depicted in Figure 9, the net effect is that a Sparcle context switch takes 14 cycles. This illustrates the total penalty for a context-switch on a data instruction. Note that, while this diagram shows 15 cycles, one of them is the fetch of the first instruction from the next context.

By maintaining a separate PC and processor status register for each context, a more aggressive processor design could switch contexts much faster. However, even with 14 cycles of overhead and four processor-resident contexts, multithreading can significantly improve system performance.^{13,14}

Support for fine-grain synchronization. As discussed earlier, fine-grain data-level synchronization is expressed with J- and L-structures and implemented using new instructions that interact with full/empty bits in memory. Sparc implements the new load, store, and swap instructions using the Sparc alternate address space instructions. We have modified these instructions in two ways:

1. The load, store, and swap alternate space instructions in Sparcle are unprivileged for ASI values in the range 0×80 to $0 \times FF$. They remain privileged for ASI values less than 0×80 . The CMMU uses the ASI value as an extended opcode; that is, ASI 0×84 corresponds to the load and

trap if empty operation. This allows user code to interact directly with full/empty bits.

2. We have used several new opcodes to produce specific ASIs on the Sparcle output pins while allowing the register + offset addressing mode. The normal load/store ASI instructions only allow register + register addressing.

A new dedicated synchronous trap line carries full/empty trap signals. J- and L-structure operations are implemented with the following special load/store instructions:

LDN	Read location
LDEN	Read location and set to empty
LDT	Read location if full, else trap
LDET	Read location and set to empty if full, else trap
STN	Write location
STFN	Write location and set to full
STT	Write location if empty, else trap
STFT	Write location and set to full if empty, else trap

In addition to possible trapping behavior, each of these instructions sets a coprocessor condition code to the state of the full/empty bit at the time the instruction starts execution. Either trapping or an explicit test of this condition code will detect a synchronization failure. When a trap occurs, the trap handling software decides what action to take.

Implementation of J-structures. To demonstrate how the special load/store instructions can be used, we will describe how we implement J-structures and present the cycle counts for various synchronizing operations. Sparcle implements a J-structure allocation by allocating a block of memory with the full/empty bit for each word set to empty. Resetting a J-structure element involves setting the full/empty bit for that element to empty. Implementing a J-structure read operation is also straightforward: it is a memory read that traps if the full/empty bit is empty. Sparcle implements it with a single instruction:

LDT (R1),R2 ; R1 points to J-structure location

If the full/empty bit is empty, the reading thread may need to suspend execution and queue itself on a wait queue associated with the empty element. To minimize memory usage, we use a single memory location to represent *both* the value of the J-structure element and the wait queue. This implies that we need to associate two bits of state with each J-structure element: whether the element is full or empty and whether the wait queue is locked or not.

```

MOVE $0, R3 ; set up swap register.
SWAPT R3, (R1) ; swap zero with J-structure location, trap if full.
CMP $-1, R3 ; check if queue is empty.
BEG, a %done ; branch if no waiters to wake up.
STFT R2, (R1) ; write value and set to full (delay slot).

:
<wake up waiters and store value>
:
%done

```

Figure 10. Machine code implementing a J-structure write.

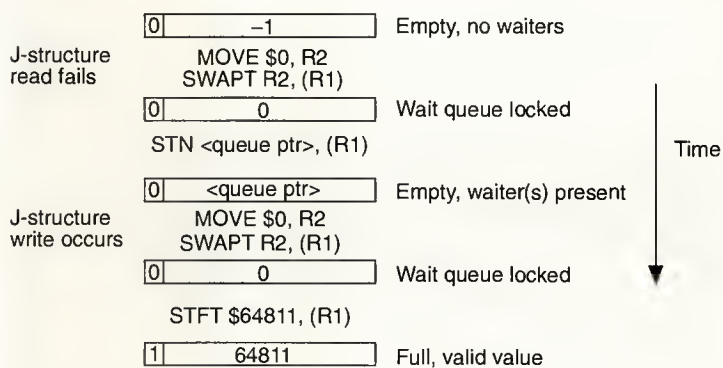


Figure 11. Reading and writing a J-structure slot.

Other architectures implement these two state bits directly in hardware by having multiple state bits per memory location.^{15,16} Instead of providing an additional hardware bit, we take advantage of Sparcle's atomic register-memory swap operation. Since the writer of a J-structure element knows that the element is empty before it does the write operation, it can use the atomic swap to synchronize access to the wait queue. With this approach, a single full/empty bit is sufficient for each J-structure element. A writer needs to check explicitly for waiters before undertaking the write operation.

Using atomic swap and full/empty bits, the machine code in Figure 10 implements a J-structure write. In this figure, R1 contains the address of the J-structure location to be written to, and R2 contains the value to be written. Also, -1 is the end of the queue marker, and 0 in an empty location means that the queue is locked. Compared with the hardware approach, this implementation costs an extra move, swap, compare, and branch to check for waiters. However, we believe that the reduction in hardware complexity is worth the extra instructions.

Figure 11 gives a scenario of accesses to a J-structure location under this implementation and illustrates the possible states of a J-structure slot. Here, R1 contains a pointer to the J-structure slot.

Table 1. Summary of fast-path costs of J-structure and L-structure operations, compared with normal array operations.

Element	Action	Instructions	Cycles
Array	Read	1	2
	Write	1	3
J-structure	Read	1	2
	Write	5	10
	Reset	1	3
L-structure	Read	2	5
	Write	5	10
	Peek	1	2

```

STIO  R2, $piout0 ; Store header.
STIO  R3, $piout1 ; Store data word.
STIO  R4, $piout2 ; Store address of data.
STIO  R5, $piout3 ; Store length of data.
IPILAUNCH 2, 1 ; Launch message. Descriptor is 2 double-
                ; words long and contains 1 double-word
                ; of explicit data (from R2 and R3).

```

Figure 12. Machine code implementing a message send.

Table 1 summarizes the instruction and cycle counts of J-structure and L-structure operations for the case where no waiting is needed on read operations and no waiters are present on write operations. In Sparcle, as in the LSI Logic Sparc, normal read operations take two cycles and normal write operations take three cycles, assuming cache hits. A locking read is considered a write and thus takes three cycles.

Support for futures and placeholders. To support futures and placeholders, Sparcle provides automatic and efficient detection and handling of placeholders via traps. Two Sparcle modifications are involved.

First, to detect placeholders, Sparcle adds two new instructions called NTADD and NTSUB. These instructions cause tag overflow traps whenever the low bit of either of their operands is set. (NTADD and NTSUB are modifications of the Sparc tagged instructions TADDCCTV and TSUBCCTV that trap whenever the low two bits of either of their operands are set.) As discussed earlier, only pointers to placeholders have the low bit set. With tag overflow traps, NTADD and NTSUB automatically detect placeholders in add, subtract, and compare operations. The address alignment trap in Sparcle detects placeholders in pointer dereferencing operations.

Second, to efficiently handle traps caused by placeholders, the trap vector number that is generated by tag overflow and

address alignment traps depends on the register containing the placeholder. This feature saves the trap handler from having to waste cycles decoding the trapping instruction to find out which register contains the offending placeholder. Johnson¹⁷ and Ungar et al.¹⁸ have proposed similar mechanisms.

Fast message handling. Most distributed shared-memory machines are built on top of an underlying message-passing substrate. Traditional shared-memory machines provide a layer of hardware that implements some coherence protocol between the processor and the interconnection network. It is natural, then, to provide the processor with direct access to the network in addition to the shared-memory interface because many operations benefit greatly from direct network access. Sparcle supports sending and receiving messages via a memory-mapped interface to the interconnection network.

Send. Sparcle sends messages through a two-phase process: first describe, then launch. Sparcle composes a message by writing directly to the interconnection network queue using a special store instruction called STIO (for store IO). The queues are memory mapped as an array of network registers in the CMMU, called the output descriptor array. In terms of performance, write operations into this array incur the same cost as write hits into the cache.

The first word of the message must be a header indicating a message opcode and the destination node. Sparcle reserves a range of opcodes for privileged use by the operating system. The rest of the message can contain immediate values from registers, or address and length pairs which invoke DMA on blocks from memory.

After the message is composed, a coprocessor instruction launches the message. Figure 12 illustrates the sending of a single message with one data word and one block of data from memory. In addition to the required header, this message includes one explicit data word and one block of data from memory. On entry to this code sequence, register R2 contains the header, R3 contains the data word, R4 the address of the data block, and R5 the length of the data block. If Sparcle is in the user mode and the header is privileged, an exception will occur. The CMMU maintains the atomicity of messages as described in the next section.

Receive. A message arrival causes a trap. The trap handler can either load words directly from the incoming message into registers using a special load instruction called LDIO (for load IO) or initiate a DMA sequence to store the message into memory. If the latter option is chosen, the processor can direct the CMMU to generate an interrupt after the storeback is complete.

Support for message handling. The following features of Sparcle support messaging:

- Special user-level load/store instructions allow fast composition of outgoing messages and fast examination of

incoming messages. An ASI value is reserved for the transferring of data to and from message register values. This ASI is produced by two new Sparcle instructions, STIO and LDIO. Although these instructions support a memory-mapped interface to the network registers, addresses for the message queues fit completely into the address offset field. Consequently, the compiler can generate instructions that perform direct register-to-register moves between the processor and the network queues.

- Register windows permit fast processing of message interrupts. One of the four hardware contexts is reserved for message processing. Consequently, the message interrupt handler needs only to alter the current window pointer so that this special context is active. No registers need to be saved and restored.
- Coprocessor instructions for message launch and disposal permit pipelining of network operations. Further, opcode bits in the launch and disposal instructions contain information about the format of messages that are about to be sent or received into memory. Thus, message format is completely under control of the compiler. Finally, the coprocessor interface permits a precise identification of the commit point for launch instructions, ensuring that message launches are atomic.
- Fast interrupt operations allow rapid entry into message handler code on the arrival of a message. In our current implementation, because interrupts always force the processor into the supervisor mode, user-level receipt of messages requires a few extra cycles for the processor to transfer control to user code. In a more aggressive implementation, the processor would support a user-level return from trap.

The CMMU interface

From this discussion we can clearly see that the Sparcle processor is part of a complete system. Consequently, several of the mechanisms that were included in Sparcle are incomplete without the support of the CMMU. Here we briefly discuss the Alewife CMMU and how it interfaces to Sparcle. Although the Alewife CMMU provides a number of features, we focus on the cache controller and message interface.

Earlier, under Alewife machine interfaces, we discussed two categories of signals in the interface between processor and CMMU: flexible data access mechanisms and flexible instruction extension mechanisms. Figure 13 makes this interface more concrete by showing Sparcle equivalent names for all of the signals. Each signal in this figure corresponds directly to signals in Figure 6.

A few of the data access mechanisms require further discussion. The modifier is implemented with the Sparc ASI field. Again, Sparcle contains a number of new load/store instructions that differ only by the values that they place on the ASI pins during data cycles. These new load/store in-

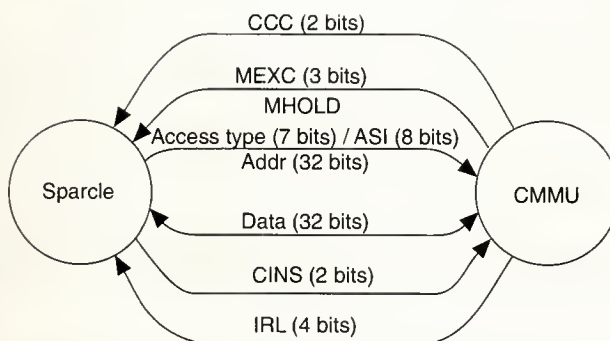


Figure 13. Sparcle signal names.

structions are important to the implementation of full/empty bit synchronization and fast messages. The trap access signals are new versions of the Sparc memory exception signal MEXC, which have distinct trap vectors. These invoke context-switch and synchronization traps. The external condition bits are implemented through the Sparc coprocessor condition codes (CCC); consequently, "branch on condition-code" instructions in Sparc can be used to examine them.

Finally, the external instruction interface is implemented directly through a Sparc coprocessor interface. Sparcle asserts one of the CINS signals to indicate that a coprocessor instruction has been decoded by the processor and should be executed by the coprocessor. Two CINS signals are required because pipeline interlocks can occasionally cause the instruction fetch unit to get ahead of the rest of the pipeline.

Latency tolerance. We already discussed rapid context switching for latency tolerance from the standpoint of the Sparcle processor. In addition to those Sparcle mechanisms, the cache controller must be able to handle multiple outstanding requests. This involves the ability to handle split-phase memory transactions (separating the request for data from the response) and to place returning data into the cache while the processor is performing some other task. Consequently, when the processor requests a data item that is not in the local cache, the cache controller asserts the appropriate trap line to initiate execution of the context-switch trap handler. At the same time, it sends a request message to the particular node that contains the requested data. Note that the mechanisms required to handle context switching differ little from those required for software prefetching. (However, see Kubiawicz, Chaiken, and Agarwal¹⁹ for some interesting forward-progress issues.)

Full/empty-bit synchronization. Full/empty-bit synchronization, as implemented in Alewife, requires support from the cache controller. Since full/empty-bit synchronization employs one synchronization bit for each data word, extra

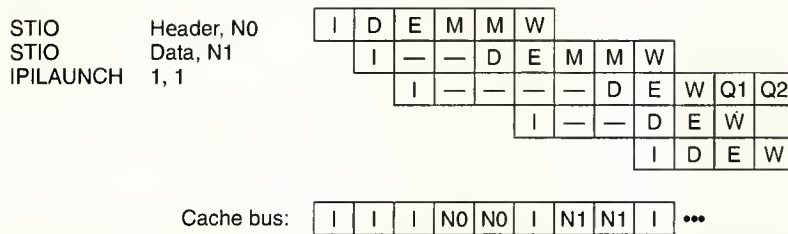


Figure 14. Pipelining for transmission of a message with a single data word.

storage must be reserved for these bits in the cache system. While these bits logically belong with the cache data, the Alewife CMMU implements them with the cache tags. This has a number of advantages. It eliminates a need for an odd number of bits in the physical memory used for cache data. It also makes access to the tags file much faster than access to the cache data, both because the tags file is smaller and because no chip crossings are required. This permits synchronization operations to occur in parallel with processing of the cache tags.

Of the Sparcle mechanisms, those important to full/empty synchronization are the external condition code, the access modifier (ASI), and one of the extra trap lines. All of the new synchronizing load/store instructions mentioned earlier are distinguished by the value of the ASI field that they generate (and whether they are read or write operations). For each data access, the Alewife CMMU takes the proffered ASI value along with the address and type of access. The CMMU uses the address to index into the tags file, retrieving both the tag and the appropriate full/empty bit. Simultaneously, it decodes the ASI value to produce two different actions, one which will be taken if the full/empty bit is full, and one if the full/empty bit is empty. When the tag lookup is completed, the CMMU completes both tags match and full/empty-bit operations simultaneously, either flagging a context-switch (on cache miss), a synchronization fault, or successful completion of the access. In all cases, the CMMU places the full/empty bit that was first retrieved from the tags file in one of the external condition codes for future examination by the processor.

The support that Alewife provides for full/empty-bit synchronization is external to the processor pipeline: that is, it occurs at the first-level cache. Consequently, full/empty bits never enter the processor core. Further, individual load/store instructions have varied semantics with respect to the full/empty bit: some cause test-and-set-like operations; others invoke traps. This places some data processing logic within the first-level cache. For modern processors that have one level of on-chip caching, a closer integration between the processor pipeline and full/empty bit synchronization might be desirable. This could include widening of internal processor registers and use of special full/empty-bit synchroniza-

tion instructions that are sandwiched between Alpha-style²⁰ load-locked/store-conditional synchronization instructions.

Fast message handling. Fast messaging in Alewife relies on a number of features in the CMMU. All of the network queuing and DMA mechanisms are a part of this chip. Sparcle interfaces with these mechanisms through both the external instruction interface and through special loads and stores. As we discussed, Sparcle reserves one special load/store instruction

(and corresponding ASI) for rapid descriptions of outgoing messages and rapid examination of incoming messages. The cache controller recognizes accesses with this ASI and causes data transfer to and from message queues instead of the cache. Message data thus transfers between the processor and network at the same speed as cached accesses.

Alewife uses the external instruction interface to implement the message launch mechanism. Consequently, message launches can be pipelined. Figure 14 gives a simple pipeline example. Here, the two-cycle latency for stores and the lack of an instruction cache limit the message throughput. More aggressive processor implementations would not suffer from this limitation. In this figure, Sparcle pipeline stages are Instruction fetch, decode, execute, memory, and writeback. Network messages are committed in the writeback stage. Stages Q1 and Q2 are network queuing cycles. The message data begins to appear in the network after stage Q2. Note that the use of DMA on message output adds additional cycles (not shown in the figure) to the network pipeline.

The close coupling between the message launch mechanism and the processor pipeline allows us to identify a precise launch completion point (corresponding to the writeback stage of the launch instruction). As a result, message launches are atomic. Before the launch instruction commits, no data is placed into the network. After the launch commits, Alewife sends a complete output packet to the network. These atomic semantics allow multiple levels of user and interrupt code to share a single network output port without requiring that the user disable interrupts before beginning to describe a message.

THE SPARCLE CHIP INCORPORATES MECHANISMS required for massively parallel systems in a Sparc RISC core. Coupled with a CMMU, Sparcle allows a fast, 14-cycle context switch, an 8-cycle user-level message send, and fine-grain full/empty-bit synchronization.

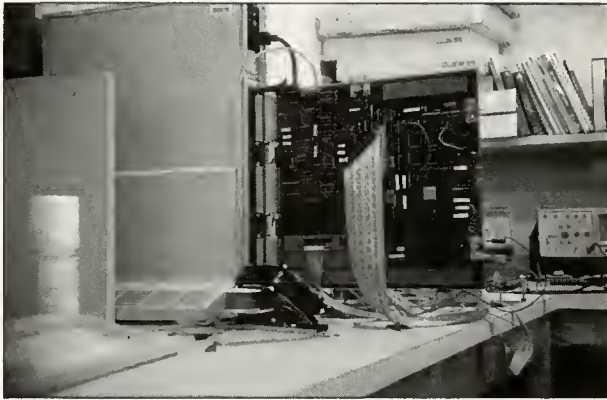


Figure 15. Sparcle's test system.

Before we received working Sparcle chips from LSI in the spring of 1992, we used an operating single-node test system. Also operational for several months was a compiler and a runtime system for our parallel versions of C and Lisp. The test system shown in Figure 15 comprises 256 Kbytes of static RAM memory, an I/O interface to the VMEbus for downloading programs and monitoring execution, and control logic to exercise the full/empty bit and context switching functionality. We had debugged the test system using Sparcs in place of Sparcles; it operated at a maximum clock frequency of about 25 MHz. (Sparc and Sparcle have only a few differing pins, and Sparcle even provides an input signal Mode pin that allows switching between Sparc and Sparcle modes.)

We have been running several parallel programs, including Sparcle's runtime system, to exercise all of Sparcle's functionality, at the maximum speed of the test bed. Scope measurements of critical signal timings on the chip's pins suggest we will be able to run the chips in an Alewife node board at roughly the same speed as the original, unmodified Sparcs.

Implementation of the Sparcle development relied on modifying an existing design through a unique collaboration with industry. Although we had our moments of trepidation, given the number of participants and the multiple failure modes (both technical and political), we believe this model of experimentation has been very successful. This implementation strategy not only allowed us, at a university, to experiment with architectural ideas in a real, contemporary processor design, it also significantly reduced the design effort from the concept stage to working chip.

Figure 16 depicts the resulting project schedule for Sparcle. We defined Sparcle's early architecture in April 1989. At MIT we also wrote a Sparcle compiler for a version of Lisp and implemented a cycle-by-cycle simulator. Later, we also developed a compiler for a parallel version of C. By March 1990, we had developed a detailed specification of the modi-

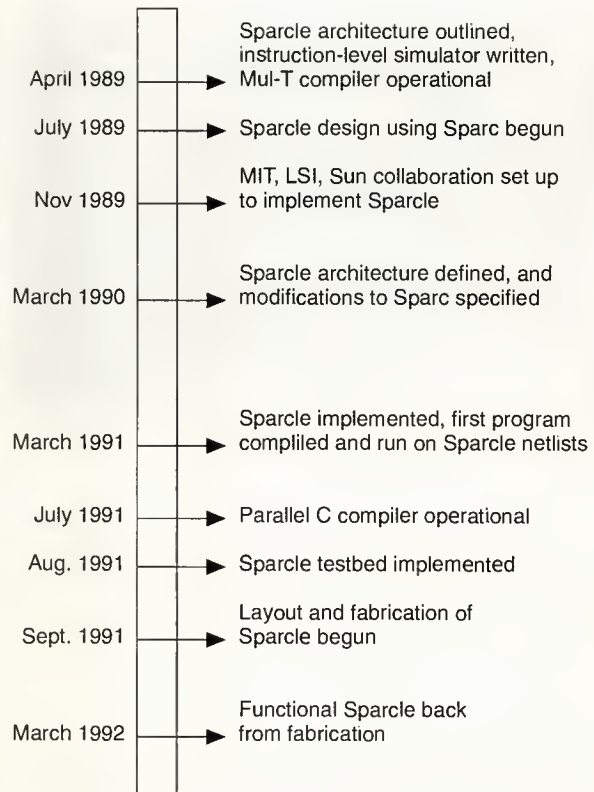


Figure 16. Sparcle's implementation schedule.

fications to Sparc required to implement Sparcle. Then, Sun made high-level changes to Sparc functional blocks, and LSI made lower gate-level changes. We tested these changes against Sparcle binaries produced at MIT. Then LSI synthesized netlists and MIT tested them against several hundred thousands of test vectors. The test vectors included both Sparc vectors provided by LSI and Sparcle vectors obtained from the MIT Sparcle simulator. The test setup included a netlist module for the floating-point coprocessor and a behavioral model for the rest of the memory and communication systems. Finally, LSI undertook layout and fabrication, during which time we also implemented a test system for Sparcle.

While the Sparcle chip project demonstrates that a contemporary RISC microprocessor can readily incorporate features considered by many to be critical for massively parallel multiprocessing, the end systems benefit of these mechanisms can only be evaluated in the context of a complete multiprocessor system. We are in the final stages of implementing the Sparcle-based Alewife multiprocessor system. Figure 1 shows an Alewife node board with the Sparcle and FPU. Figure 17 shows a 16-node Alewife system package developed by the Advanced Production Technology group



Figure 17. The 16-node Alewife package.

at the Information Sciences Institute in Los Angeles. The CMMU chip has been implemented and tested. It is being implemented in LSI Logic's LEA 300K process, and we expect to begin its fabrication shortly. ■

Acknowledgments

The Sparcle project is funded in part by DARPA contract N00014-87-K-0825 and in part by National Science Foundation grant MIP-9012773. LSI Logic and Sun Microsystems helped implement Sparcle, and LSI Logic supported the fabrication of Sparcle. We acknowledge the contributions of Dan Nussbaum, who was partly responsible for the processor simulator and runtime system, and was the source of several ideas. Halstead's work on multithreaded processors influenced our design. Our research also benefited significantly from discussions with Bert Halstead, Tom Knight, Greg Papadopoulos, Juan Loaiza, Bill Dally, Steve Ward, Rishiyur Nikhil, Arvind, and John Hennessy.

References

1. *Sparc Architecture Manual*, Sun Microsystems, Mountain View, Calif., 1988.
2. A. Agarwal et al., "The MIT Alewife Machine: A Large-Scale Distributed-Memory Multiprocessor," *Proc. Workshop on Scalable Shared Memory Multiprocessors*, Kluwer Academic Publishers, Boston, 1991. An extended version of this paper has been submitted for publication and appears as MIT/LCS Memo TM-454, 1991.
3. A. Agarwal et al., "APRIL: A Processor Architecture for Multiprocessing," *Proc. 17th Ann. Int'l Symp. Computer Architecture*, IEEE CS Press, Los Alamitos, Calif., June 1990, pp. 104-114.
4. B.J. Smith, "Architecture and Applications of the HEP Multiprocessor Computer System," *Soc. of Photooptical Instrumentation Engineers*, Bellingham, Wash., Vol. 298, 1981, pp. 241-248.
5. Arvind, R.S. Nikhil, and K.K. Pingali, "I-Structures: Data Structures for Parallel Computing," *Trans. on Programming Languages and Systems*, Vol. 11, No. 4, Oct. 1989, ACM Press, pp. 598-632.
6. M. Dubois, C. Scheurich, and F.A. Briggs, "Synchronization, Coherence, and Event Ordering in Multiprocessors," *Computer*, Vol. 21, No. 2, Feb. 1988, pp. 9-21.
7. S. V. Adve and M.D. Hill, "Weak Ordering—A New Definition," *Proc. 17th Ann. Int'l Symp. Computer Architecture*, IEEE CS Press, June 1990, pp. 2-14.
8. D. Lenoski et al., "The Directory-Based Cache Coherence Protocol for the DASH Multiprocessor," *Proc. 17th Ann. Int'l Symp. Computer Architecture*, IEEE CS Press, June 1990, pp. 148-159.
9. D. Kranz et al., "Integrating Message-Passing and Shared-Memory; Early Experience," to be published in *Conf. Principles and Practice of Parallel Programming*, ACM, May 1993, and appears as MIT/LCS Memo TM-478, 1993.
10. D. Chaiken, J. Kubiawicz, and A. Agarwal, "LimitLESS Directories: A Scalable Cache Coherence Scheme," *Proc. Fourth Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS IV)*, ACM, Apr. 1991, pp. 224-234.
11. W.J. Dally et al., "The J-Machine: A Fine-Grain Concurrent Computer," *Proc. Int'l Federation for Information Processing (IFIP) 11th World Congress*, Elsevier Scientific Publishing, New York, 1989, pp. 1147-1153.
12. C.L. Seitz et al., "The Design of the Caltech Mosaic C Multicomputer," *Proc. 1993 Symp. Research on Integrated Systems*, G. Borriello and C. Ebeling, eds., MIT Press, Cambridge, Mass., 1993, pp. 1-22.
13. W-D. Weber and A. Gupta, "Exploring the Benefits of Multiple Hardware Contexts in a Multiprocessor Architecture: Preliminary Results," *Proc. 16th Ann. Int'l Symp. Computer Architecture*, IEEE CS Press, June 1989, pp. 273-280.
14. K. Kurihara, D. Chaiken, and A. Agarwal, "Latency Tolerance Through Multithreading in Large-Scale Multiprocessors," *Proc. Int'l Symp. Shared Memory Multiprocessing*, IPS Press, Japan, Apr. 1991, pp. 91-101.
15. G. Alverson, R. Alverson, and D. Callahan, "Exploiting Heterogeneous Parallelism on a Multithreaded Multiprocessor," *Workshop on Multithreaded Computers, Proc. Supercomputing*, ACM Siggraph, Nov. 1991.
16. G.M. Papadopoulos and D.E. Culler, "Monsoon: An Explicit Token-Store Architecture," *Proc. 17th Ann. Int'l Symp. Computer Architecture*, IEEE CS Press, June 1990, pp. 82-91.
17. D. Johnson, "Trap Architectures for Lisp Systems," *Proc. 1990 ACM Conf. on Lisp and Functional Programming*, 1990, pp. 79-96.
18. D. Ungar et al., "Architecture of SOAR: Smalltalk on a RISC," *Proc. 1984 Int'l Symp. on Computer Architecture*, 1984, pp. 188-197.
19. J. Kubiawicz, D. Chaiken, and A. Agarwal, "Closing the Window

of Vulnerability in Multiphase Memory Transactions," *Fifth Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS V)*. ACM, Oct. 1992, pp. 274-284.

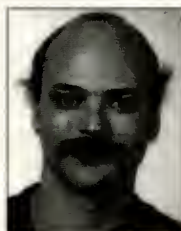
20. *Alpha Architecture Reference Manual*, Digital Press, Maynard, Mass., 1992.



Anant Agarwal serves at the Laboratory for Computer Science at MIT where he is an associate professor of electrical engineering and computer science. His current research interests include the design of scalable multiprocessor systems, VLSI processors, compilation and runtime technologies for parallel processing, and performance evaluation.

At Stanford University, he participated in the MIPS and MIPS-X projects. He initiated the Alewife project at MIT, which is aimed at the design and implementation of a large-scale cache-coherent multiprocessor.

Agarwal received a B Tech in electrical engineering from the Indian Institute of Technology, Madras, India, and an MS and PhD in electrical engineering from Stanford. He is a member of the ACM and IEEE Computer Society.



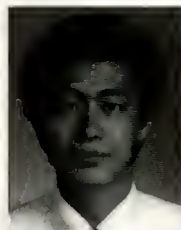
John Kubiawicz is a doctoral candidate in the Department of Electrical and Computer Science at MIT. His current research interests include parallel computer architecture, high-performance microprocessor design, and high-energy particle physics.

Kubiawicz received BS degrees in electrical engineering and physics and an MS in electrical engineering from MIT.



David Kranz has been a research associate in the MIT Laboratory for Computer Science since 1987. His research interests are in programming language design and implementation for parallel computing.

Kranz received a BA from Swarthmore. While earning a PhD at Yale, he worked on high-performance compilers for Scheme and applicative languages.



Beng-Hong Lim is currently a doctoral candidate in MIT's Department of Electrical Engineering and Computer Science. His research interests include parallel computing and computer architecture. He received a BS and an MS in electrical engineering and computer science from MIT.



Donald Yeung is a PhD candidate at MIT. His research interests are in the area of multiprocessor design, including efficient hardware and software mechanisms for synchronization and latency tolerance.

Yeung received a BS from Stanford in computer systems engineering, and recently completed an MS in electrical engineering and computer science at MIT.



Godfrey D'Souza has been with the Spare Systems Division and the CoreWare Group at LSI Logic where he is a senior design engineer involved with aspects of microprocessor and system level design.

D'Souza received a BS in electronics and communications engineering from the University of Baroda, India, and an MS in electrical engineering from the University of Washington, Seattle. He is a member of the IEEE Computer Society.



Mike Parkin is a staff engineer at Sun Microsystems, where he is currently part of a research team that is building a Sparc-based scalable multiprocessor system.

Parkin received a BSEE from Iowa State University and an MSEE from Stanford.

Direct questions concerning this article to David Kranz, MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139, or via e-mail at kranz@lcs.mit.edu.

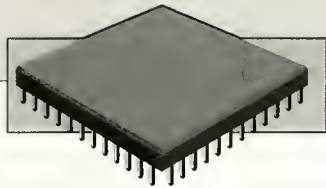
Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 162

Medium 163

High 164



Message-Routing Systems for Transputer-Based Multicomputers

An efficient message-routing system, an essential component of a multicomputer, allows communication between any two processes of a concurrent program wherever they are located on a parallel computer network. It simplifies the concurrent software development on multicomputers by separating the hardware architecture from the software configuration of processes. This article surveys some implemented routers for transputer networks and compares them for adaptivity, deadlock freedom, network latency, generality, and livelock freedom.

Domenico Talia

CRAI

Solving the communication problems inherent in a multicomputer composed of a large number of computing elements requires efficient data handling and interconnections. The processor in each computing element is tightly coupled to a local memory that is physically separated from the memories of other computing elements (Figure 1). Thus, on a multicomputer a message-routing system is necessary to control the passing of data between different processors. It lets us simplify the concurrent software development on parallel computers by separating the hardware architecture from the software configuration of processes.

One solution involves the single-chip Inmos transputer. This complete microcomputer contains communication links that interconnect to other transputers to form multicomputer systems. In particular, the T400 and T800 generations of transputers support communication among processes running on processors directly connected by a physical link. However, many parallel algorithms require a process running on one processor to communicate with processes running on other processors not directly connected by a link. To implement these algorithms on transputer-based multicomputers, we must include a through-routing system.

Recently, designers have introduced several message-routing systems for transputer-based

multicomputers for direct use by a programmer.¹⁻⁶ In addition, some communication facilities have been provided in developing environments such as Express, Trollius, and Helios. Many of the routing systems are software implementations of routing algorithms that are defined for more general network topologies. Two of these, Interval Labelling³ and Mad Postman,⁵ will be implemented in hardware by a routing chip that connects to transputers and handles the message traffic.

Inmos is implementing the Interval Labelling routing in its IMS C104 communication device, which will enable communications among T9000 transputers. The MP1 network chip will incorporate the Mad Postman routing strategy proposed by Yantchev and Jesshope.⁷ This device is not committed to interface only to the transputer; in fact, an interface is required between the processor that uses the MP1 chip to route messages. Two interfaces to the transputer have been developed. To date, neither the IMS C104 nor the MP1 is commercially available.

Another solution to communication problems on transputer networks is the link-switching network. In this approach, a set of switched links provides a highly connected, virtual network. Several link switch devices have been implemented and are currently in use. The best known is the Inmos C004 crossbar switch, which can switch

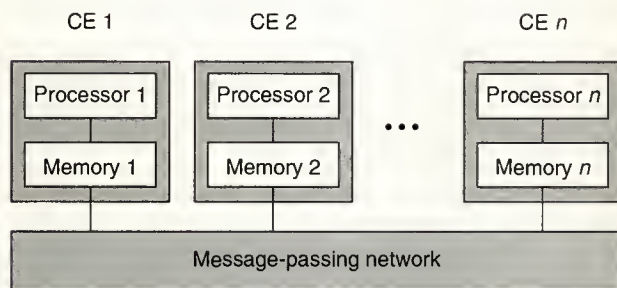


Figure 1. Multicomputer architecture.

32 input links to 32 output links with 20-Mbit/s operating speed. Other link switches support the Meiko Computing Surface, in the Floating-Point Systems T-series machines, and in other commercial transputer-based multicomputers.

In particular, the ESPRIT Supernode Project has implemented a link switch system for the Telmat T.Node system.⁸ In it a pair of switch chips can switch 72 input links to 72 output links. Thus, four pairs of switch chips can switch 72 transputers into any link configuration.

With switched networks messages sent between two nodes must also pass through link switches, incurring an additional transmission delay. A user-required topology can be set on the network, but it is not a real alternative to through routing if dynamic link reconfiguration is not enabled. However, dynamic link reconfiguration requires an overhead due to the time taken for satisfying link switch requests.

Here, I survey some implemented routing systems for transputer networks and compare them with respect to several criteria, such as adaptivity, deadlock freedom, generality, livelock freedom, and network latency.

Except for one, these message-routing systems support the programming of concurrent applications on transputer networks. They present a user view of a transputer network as if all the transputers were completely connected. The message-routing systems are Tiny,¹ CSN,⁴ Multiple Rings,² and Ordered Dimensions.⁶ Finally, for its primary importance in future transputer-based multicomputers, I also describe the Interval Labelling routing that will be used in the latest generation T9000 transputer.

Message routing

As mentioned before, in a multicomputer system composed of a set of computing nodes connected by a communication network without shared memory, one of the main problems that must be faced is the routing of messages between the nodes. (See the Routing algorithms box.)

To be a practical communication support for programs running on multicomputers, a routing system must offer the following major features:

Routing algorithms

A routing algorithm determines a path between two nodes that are not directly connected. That is, it determines which output channel on a node i must route a message directed to a process running on a remote node j , as shown in Figure A. More formally, a routing algorithm is a *routing function* $R: N \times N \rightarrow C$ that maps the current node n_c and destination node n_d to the channel c_n on the route from n_c to n_d , $R(n_c, n_d) = c_n$.

Routing algorithms can be divided into two main classes, *adaptive* and *oblivious*. In adaptive algorithms the routing paths are established dynamically, whereas in oblivious algorithms they are statically defined. *Deterministic* routing is a special case of oblivious algorithms in which a single route exists between two nodes. In fact, in deterministic algorithms, the route followed by a message sent from node i to node j is predetermined by its source-destination pair (i, j) . For example, the communication systems of the Symult series 2010 and the Intel iPSC/2 employ an oblivious routing technique.

Another way to classify routing algorithms is based on the policy used to propagate a message from node to node. In *store-and-forward* routing a message is first entirely stored in each node on the path and then it is transmitted to the next node.

Different techniques are *cut-through* and *wormhole* routing. According to these techniques a message is broken down in *flits*, the smallest unit of information that a channel can accept or refuse. Instead of storing a message in a node and then transmitting it to the next node, the wormhole and cut-through techniques operate by advancing the head of a message directly from an incoming to an outgoing channel. Only a few flits are buffered at each node. The first flit (the *head*) holds the destination's address. Once a link

is occupied by the head, it cannot be used for other messages until the last flit of the message has left it. With wormhole routing, blocked messages remain in the network. Virtual cut-through differs from wormhole routing in that it buffers messages when they block, removing them from the network.

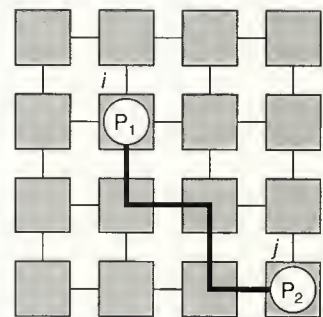


Figure A. A path between two processes.

Definitions

Adaptivity: The ability to choose the message route depending on traffic load and network topology. In adaptive algorithms the routing function is based on measurements or estimates of message traffic and network configuration.

Deadlock: A condition that occurs in a communication network when no message is buffered at its destination and no message can advance because all buffers in each routing path are full.

Deadlock freedom: The ability to avoid or recover from a deadlock occurrence.

Generality: The ability of a message-routing system to be used on one or more classes of network topologies.

Livelock freedom: A property that guarantees a message will be delivered to its destination in a finite time.

Network latency: The time it takes a message to leave the source and reach its destination.

- the routing system must be free of deadlocks;
- the message latency must be low; and
- no message may be infinitely delayed in the network.

Other relevant features are adaptivity and generality. (See Definitions box.)

Deadlock freedom. Deadlock is a typical problem of distributed computing. Since every distributed algorithm must solve a deadlock situation, to be practical, a message-routing system must be properly designed to avoid this occurrence.

Generally, in routing algorithms deadlock is related to the presence of cyclic paths in the network. Several techniques implement deadlock freedom in routing systems. For example, a simple policy may discard a preempted message. In this case the source node must be notified, and a retransmission mechanism must be provided.

An interesting technique to avoid deadlock is based on an ordering of the network channels.⁹ Physical channels belonging to cycles are split into a group of virtual channels. The virtual channels are ordered, and message routing is restricted to visit channels in decreasing order to eliminate cycles.

Network latency. In store-and-forward routing the network latency (T_l) is the product of the mean internode latency (hop time) and the average distance (number of hops):

$$T_l = T_{h1} N_h$$

where T_{h1} is the hop time and N_h is the number of hops. In wormhole routing the network latency is

$$T_l = T_{h1} N_h + (K/B)$$

where T_{h1} is the routing delay in each node for sending the message head to its destination. N_h is the number of hops, and K/B is the time required to move the whole message K (bytes) through the wormhole channel of bandwidth B (bytes/s).

Since high communication costs may abate the benefits deriving from the parallel execution of programs, a practical routing system should have low mean latency to deliver messages to their destinations.

Livelock freedom. Livelock occurs when a message never arrives at its destination. Oblivious routing avoids livelock if the queue buffering is fair and the paths are minimal. In adaptive routing this is not sufficient; generally, a priority must be assigned to each message when it is sent. This priority will be increased as the message remains in the network, and messages are routed respecting their priority.

Adaptivity. Although an adaptive algorithm may be restricted to a particular topology, it might achieve a better performance than an oblivious one. The system can adapt to traffic conditions and choose the communication paths, avoiding hot spots. Moreover, adaptive algorithms help programmers increase their productivity because they do not need to analyze network hot spots at coding time.

Generality. While some routing systems work in one topology class (such as hypercubes, meshes, trees), some other systems support a particular network. A routing system for transputer-based multicomputers can be considered as general if it runs on the topologies that can be built using the four physical links: mesh, ring, tree.

Routing systems

The five routing systems for transputer networks are Tiny, Multiple Rings, CSN, Ordered Dimensions, and Interval Labelling. Except for Interval Labelling, which will be used with T9000 transputers, these systems actually support through-routing in concurrent applications implemented on transputer-based multicomputers.

Tiny. Tiny is a message-routing harness developed for T800 transputers at Edinburgh Parallel Computing Centre.¹ In Tiny the message routing is based on routing tables recording the paths between any two processors in the network. At starting time Tiny explores the network topology and builds routing tables, storing the shortest paths (one or more) from each processor to each other.

Tiny implements two point-to-point routing strategies, sequential and adaptive, and a broadcast (one-to-every-other) strategy. Users may select the two point-to-point routing strategies at initialization time. Every strategy uses the store-and-forward technique to propagate messages.

The sequential strategy is implemented by using the same shortest path from any through-routing processor toward the destination. This strategy provides provable deadlock free-

dom by eliminating cycles in routing tables when these are built. This is achieved by routing a message to a destination through different links on a processor, depending on the link through which that message reached that processor. A useful feature of the sequential strategy is that it guarantees messages will arrive at their destination in the order in which they were sent.

The adaptive strategy determines at each through-routing processor which of the shortest paths from that processor to the destination is the best. To make the decision, Tiny examines the output queues of the local processor's appropriate links and enqueues the message to the link with the shortest queue.

To implement broadcast communications, Tiny uses a third strategy. A message broadcast by a process is received directly by all the other processes in the network instead of being restricted to only one process. The broadcast strategy uses a broadcast tree determined during initialization for each processor P_i . Information is stored in every other processor P_j to indicate its relative position in the P_i tree. Figure 2 shows a broadcast tree from node P_1 .

Tiny is a communication kernel running on each transputer and connected to one or more client processes by channels. Several agents (processes implemented in C language and transputer assembler) on each transputer implement Tiny. Each process handles an external link or a channel connecting Tiny to a user process. Tiny offers to a user an interface that provides a set of communication primitives implementing read and write operations from and to any other process in the network:

```
void pktRead (CHAN *in, int *source, *message, size)
void pktWrite (CHAN *out, int *dest, *message, size).
```

Tiny also provides a broadcast communication function:

```
void pktBroadcast (CHAN *out, int *message, size).
```

Some experiments have measured Tiny's performance in routing messages on a network of T800 transputers with 20 Mbits/s links. The internode latency (one hop) for 64-byte messages is about 200 microseconds and for 256-byte messages about 500 μ s with a low load in the network.

Multiple Rings. CRAI (Consorzio per la Ricerca e le Applicazioni di Informatica)² designed and implemented this routing system. It is based on a deadlock-free adaptive routing algorithm for k -ary n -cubes, which extends Roscoe's proposal for ring topologies.¹⁰ The transputer implementation represents a two-dimensional version of the algorithm. Four defined interconnected rings may transfer messages between rings according to the shortest path and the network load.

Messages transfer from node to node using the store-and-forward technique. In this routing system deadlock is avoided,

assuring that on each ring a node outputting a message must be prepared to accept a message from that ring. To allow rerouting from one ring to another, this system always maintains one free buffer for input for each ring when ring input is required. Thus, the buffer space required by the router to guarantee deadlock freedom is very small, and it is independent from the network dimension (that is, number of nodes).

Multiple Rings uses four rings connecting all the transputers that compose the two-dimensional torus. Two of the rings run along the columns and two along the rows, with the message traveling in opposite directions on each of the row and column rings (Figure 3). The design of the routing system incorporates these criteria:

- after a message has been output on a ring, the node will be prepared to receive an input message on that ring;
- a message can be transferred between rings only if the node has adequate buffer space; and
- the binding of a message to a node's output link need not occur until that output link becomes available.

Multiple Rings is adaptive. Once a message is input from a ring, the system generates three possible rings on which the message might be output: PreferredRing, OptionalRing, and RequiredRing. The first two rings optimally reduce the Cartesian distance; the third one is the ring on which the message is received. If possible, the routing occurs by choosing the ring for output that

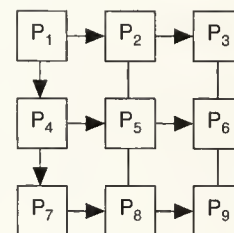


Figure 2. A broadcast tree from P_1 .

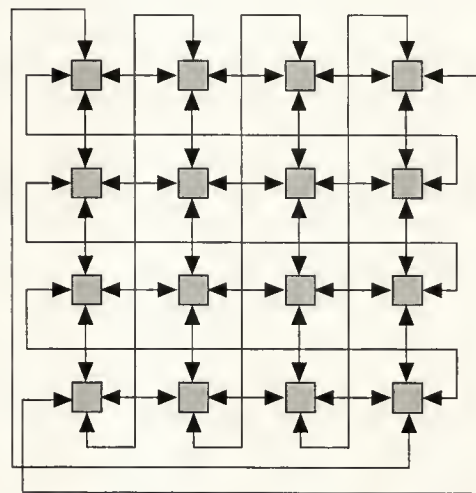


Figure 3. Two-dimensional toroidal topology.

***Multiple Rings applies to all of
the computer networks in which
at least one Hamiltonian circuit
can be identified.***

reduces the Cartesian x,y mesh distance from the current node to the node to which the message is addressed. This routing does not guarantee that the message will ultimately arrive at its destination. If the message fails to arrive at its destination within some interval, routing relative to the rings, not Cartesian coordinates, begins.

To identify when the routing should change, each message contains an initial jump count. When the message's jump count reaches zero, the message transfers between rings, only to reduce the ring distance to the target node. If the message can transfer to a ring that reduces the distance to the destination, the message will reach the target node on this ring. Otherwise the message will flow on the original ring until reaching the target node. This approach guarantees livelock freedom. It also tends to reduce the local congestion because some of the messages will be routed with a different strategy.

The system has been implemented in the Occam 2 language. The algorithm is scalable in terms of managed rings and then can exploit the overall network bandwidth. Note that this routing system applies to all of the computer networks in which at least one Hamiltonian circuit can be identified.

One source presents a performance study of the system for a network of T800 transputers with 10-Mbit/s links.² Other experiments used links with a 20-Mbit/s bandwidth. In this case the mean internode latency increases with demand from 250 to 600 μ s for 64-byte messages and from 450 to 1,200 μ s for 256-byte messages.

CSN. The Computing Surface Network communication system developed at Meiko Scientific Ltd.⁴ provides full through-routing support for concurrent programs running on the Meiko Computing Surface.

The Computing Surface is a multiuser, parallel computer based on a mix of T800 transputers, Sparcs, and i860 processors. In fact a single large Computing Surface can be viewed as a number of independent machines defining a number of user domains. Each such domain contains a set of processing elements that are accessible to the user. A generic processing element has four main components: compute processor, memory system, communication engine, and control interface. The communication engine comprises one or more transputers, which implement the CSN system.

CSN implements communication between two or more processes by means of an interconnection entity called a transport. It is an asymmetrical and bidirectional way of communication that has one owner process. A transport is a service point to which a process may send a message and from which it may receive a message without specifying the source.

The basic idea of this communication system is similar to that of other conventional communication mechanisms, such as Unix Streams. Message exchange among processes is not carried out through channels, as in the other routing systems discussed here, but by means of network access points, the transports, which might be used by many processes. A transport can be created dynamically from a process, and after its creation it can be used in each direction (send or receive) for data communication with a dynamic set of partners.

For transport management, CSN provides a distributed name service called directory server. The directory server dynamically associates a network physical address to each transport name. When a transport is created, its identifier includes three fields:

- *A global name.* This unique name is defined by the owner and must be used by partners.
- *A local name.* This is the name of the transport used in the owner process to send and receive through that transport.
- *A network address.* This is a physical address associated with the transport in the processor network. To this address is associated a buffer on which all the messages exchanged using that transport are enqueued.

When a process tries to receive a message, it does not need to specify the name of the sender; it is sufficient to use the local name of the transport on which the message will be read. Two primitives are provided to send and receive data:

`csn_tx(Trans, ..., Msg)` and
`csn_rx(Trans, ..., Msg).`

Trans is the transport name; Msg is the message to be exchanged. The communications can be synchronous or asynchronous and specify a different parameter in the primitives.

As in the other routing systems, data exchanges through a network of processes located on each transputer of the Computing Surface. For processor-to-processor message transmission, CSN uses a technique similar to virtual cut-through, dividing messages into units of 32 bytes. Even if a message smaller than 32 bytes is sent from an application's process, a 32-byte packet will travel on CSN.

Ciampolini, Corradi, and Leonardi¹¹ describe some experimental experiences with transports and present some measurement of message latency on a Meiko Computing Surface. The experiment used nine transputers connected in a mesh

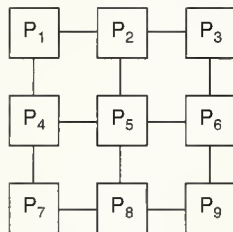


Figure 4. Two-dimensional mesh network.

topology and eight processes; each process ran on a different transputer and sent messages toward a receiver located on the ninth transputer. The latency to send 10,000 medium-size messages ranges from 10 to 60 seconds, depending on the transmission frequency. Taking into account that for this network the mean distance is 2.25, we obtain an internode latency ranging from 450 to 2,600 μ s for medium-size messages.

Ordered Dimensions. The Inmos Central Application Group in Colorado Springs, Colorado, developed this message-routing system,⁶ basing it on the ordered dimensions approach proposed by Dally and Seitz.⁹

This approach partitions the network channels into classes and orders them within classes in a way that reflects the topology of the network. Deadlock freedom derives from the fact that the channel structure does not contain cycles. Cycles are broken by adding virtual channels to the network. Several logically independent virtual channels might be mapped on a physical link.

Channel classes partition the unordered set of channels in a network. Each class C is an ordered set of channels. An ordered set of channels $C = (c_1, c_2, \dots, c_n)$ defines a channel sequence that a message can follow from the source to its destination ($c_1 < c_2 < \dots < c_n$). Classes are themselves ordered to impose a dependency on one another. Thus a traveling message having traversed the channels of one class will not revisit them after traveling through channels of the next class. Classes often correspond to dimensions of the network, for example, the x and y dimensions of a two-dimensional mesh.

As an example, the routing algorithm for a two-dimensional mesh (Figure 4) first defines two ordered dimension classes: the Y dimension consisting of north and south channels and the X dimension consisting of east and west channels. A message must complete its routing in the X dimension before traveling in the Y dimension. Each dimension class further consists of two independent direction classes. A message travels east or west, but never in both directions. Channels are ordered within each direction class.

Buffers are provided to each channel class. Additional internal channels allow switching from the higher order X di-

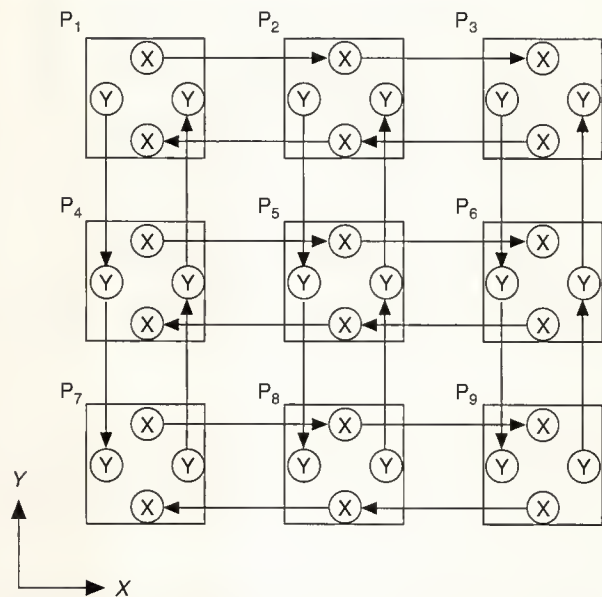


Figure 5. Logical network.

mension to the lower order Y dimension. The resulting logical network is shown in Figure 5.

To route messages in this network, the algorithm compares the current Cartesian coordinates with that of the destination. The message will move along the east/west direction until the correspondent coordinate matches the destination. Then the algorithm switches to the Y dimension (north/south direction), and the procedure repeats until the destination is reached.

This routing system has been implemented for a three-dimensional hypercube and for a two-dimensional mesh of transputers. For node-to-node message transmission the system uses the store-and-forward technique. Shumway⁶ describes the Occam 2 source code of the two routing algorithms.

Ordered Dimensions routing can be applied to any regular topology such as hypercubes, tori, meshes, and rings. The message routing is oblivious. In fact, the path taken by a message is statically defined because it is derived from the channel dependency graph. Thus a network cannot adapt its routing to the message traffic. Furthermore, the ordering over all the channels allows only one path between two nodes. Although this condition can lead the network to congestion, it may somewhat benefit a user because for each pair of nodes, messages are received in the order they were sent.

Another critical aspect of this system is that livelock freedom is not implemented. Guaranteeing livelock freedom requires provision of a fair scheduling of competing channels. However, the system does not use this kind of scheduling

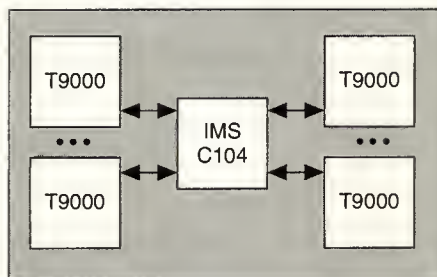


Figure 6. T9000 and IMS C104 connection.

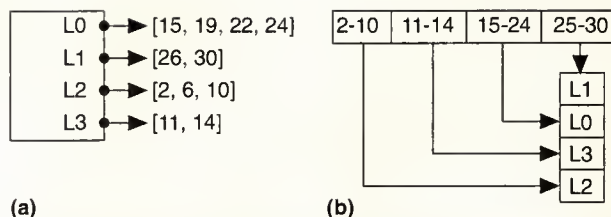
because it actually reduces the system performance.

The internode latency of the system on a T800 mesh topology increases with demand from about 100 to 300 μ s with a message size of 64 bytes, and from 210 to 700 μ s with a message size of 256 bytes.

Interval Labelling. Inmos is implementing this routing scheme in hardware. The IMS C104 communication support device for the T9000 transputer includes a full 32×32 crossbar switch, enabling messages to be routed from any of its links to any other link. One or more T9000s might be connected to a IMS C104 device, which enables the connection with other transputers in the network (Figure 6). In particular, a single C104 can provide full connectivity between 32 T9000s.

The Inmos T9000 is the latest generation transputer. The main goal for the T9000 was to improve the performance of the transputer while maintaining compatibility with existing transputer products. The T9000 provides an order-of-magnitude increase in performance with respect to the T800 and implements the same instruction set as the existing T805 transputer. In the T9000 transputer, designers used CMOS technology to integrate a 64-bit floating-point processor, a 32-bit integer processor, 16 Kbytes of cache memory, a virtual channel processor, and four high-bandwidth links on a single chip.

The key features of the T9000 architecture are a pipelined superscalar CPU combined with 16-Kbyte on-chip RAM and improved communications (80 Mbyte/s bidirectional bandwidth) to make multiprocessor programming easier. The pipelined superscalar architecture executes several instructions on each clock cycle and operates at a clock speed of 50 MHz. Instructions execute in a five-stage pipeline. The first stage can fetch two local variables; the second can execute two address calculations for accessing variables; the third can load two nonlocal variables; the fourth can perform an FPU or ALU operation; and the final stage can perform a write or conditional jump. The T9000 also provides a grouper to assemble groups of instructions. One group can be sent through the pipeline every cycle to make the best use of the hardware.



(a)

(b)

Figure 7. Interval Labelling: destinations (a) and table (b).

The major goal of achieving a significant performance increase produced a design with a peak performance in excess of 200 MIPS and 25 Mflops and a sustained performance exceeding 80 MIPS and 10 Mflops.

The communication facilities of the T9000 have been extended in comparison to the T800. On previous transputers the user was limited to assigning two software channels, one in each direction, to each hardware link. In the T9000 new multiplexing hardware allows the mapping of any number of channels on each physical link between two directly connected transputers. A communication processor called the virtual channel processor (VCP) handles these software channels. Whereas the VCP allows communication between two directly connected T9000 transputers, the C104 chip provides hardware support for routing messages across a network of T9000s.

With the T9000 and the IMS C104, we can define communication channels between two processes independently of where they are physically located or whether the channels are routed through the network. Each link of this network should support a bidirectional bandwidth of about 150 Mbps.

The routing algorithm used in the IMS C104 is called Interval Labelling.¹² This scheme assigns a distinct label to each transputer in a network. On each IMS C104, each output link has an associated interval (set of consecutive labels). The intervals associated with the links are not overlapping. As a message arrives, the algorithm examines the address to determine which interval contains a matching label then forwards the message along the associated output link.

Figure 7 shows an example of Interval Labelling in which a set of reachable destinations is defined for each link. According to the table, a message with destination address 22 ($15 < 22 < 24$) is routed through link 0 (L0).

Any network topology can be labeled so that the routing is deadlock free. This sometimes produces a nonoptimal routing that cannot exploit all of the links in the network, such as for ring topologies. On the other hand, optimal deadlock-free labelling is possible for trees, meshes, hypercubes, and multistage networks. These labellings will be provided with the routing system.

As an example, we show how to label a network with a

mesh topology. An N -dimensional mesh is composed of M meshes of dimension $N - 1$, with M corresponding nodes, one from each $N - 1$ -dimensional mesh, joined to form a line. If each of the $N - 1$ -dimensional meshes has P nodes, these are numbered $0, \dots, P - 1; P, \dots, 2P - 1; \dots; (M - 1)P, \dots, MP - 1$. In this way a mesh may be labeled to route messages in a deadlock-free manner.

Figure 8 shows an example of a simple network composed of four T9000 transputers and three C104s. The interval notation $[2,4)$ should be read as meaning that the label value must be greater than or equal to 2 and less than 4 to lie within this interval. If a message with label 3 is sent from T0, it passes through the three C104s before going to the T3 transputer.

Interval Labelling uses a wormhole technique in which the routing decision occurs as soon as the head of the message has been received. With a network of IMS C104s wormhole routing does not affect through-routing transputers, minimizing the network latency in the message transmission. Moreover, this routing system ensures livelock freedom.

The Interval Labelling routing is oblivious; in fact, it is not able to adapt the message routing according to the communication load of the network. This aspect may create an excessive amount of communication on some link that will become a hot spot in the network. To eliminate hot spots, the IMS C104 should optionally implement a two-phase routing algorithm in which each message is first sent to a random intermediate destination, then on to its final destination.

To implement this two-phase strategy, the system must prepend each message with a random header that indicates the intermediate address. To be sure that deadlock does not occur, the two phases must use separate links. This is possible by assigning random headers and destination headers from distinct intervals. Thus the interval associated with a given link on an IMS C104 must be a subinterval of the ran-

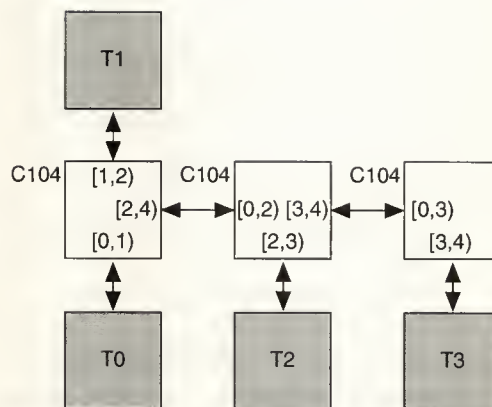


Figure 8. Interval Labelling of a simple network.

dom or destination header range. Deadlock freedom is also assured. However, the two-phase routing algorithm introduces an additional communication overhead.

No experimental measures were available when this article was written. Some simulation results give the mean internode latency at about 2 or 3 μ s. If these results are confirmed in the real use of the IMS C104 chip, the Interval Labelling routing system will represent a very good solution for message routing in the next generation of transputer-based multicomputers.

Comparisons

The message-routing systems discussed here can be effectively used to support network communications in parallel programs running on multi-transputer systems. They have different features, which can be compared for a better evaluation. Table 1 summarizes the main features of the routing systems.

Table 1. Summary of main features of the routing systems.

Systems	Transmission	Adaptivity	Deadlock freedom	Livelock freedom	Generality
Tiny	Store-and-forward	Yes No	No Yes	No Yes	All
Multiple Rings	Store-and-forward	Yes	Yes	Yes	All topologies with at least one Ham. circuit
CSN	Virtual cut-through	No	Yes	Yes	All
Ordered Dimensions	Store-and-forward	No	Yes	No	Any regular topology
Interval Labelling	Wormhole	No	Yes	Yes	All

***Generally, the routing systems
discussed here outperform the
message-passing systems
provided with the operating
systems developed for
transputer-based machines.***

These routing systems are not used as part of an operating system. Parallel programs written using these systems can be linked and will work without changes on many hardware topologies. There is no unanimous answer to whether the message-routing facility should be in the operating system or, as in the systems discussed here, available without any higher abstraction level. While the integration in an operating system, such as Helios and Trollius, has its advantages—both because they offer a standard interface and often high-level primitives—it imposes high software overheads. Since the communication overhead is a very critical issue for most parallel programs running on multicomputers, many users prefer to use a message-routing system.

Deadlock freedom. As mentioned before, a system unable to avoid deadlock occurrence is not practical. All the routing systems surveyed here are deadlock free. The only exception is represented by the adaptive strategy that can be optionally used in Tiny. This routing strategy is not practical because cycles can be created in a message route so deadlock can occur.

The routing systems use different techniques to avoid deadlock. The approach used in CSN and Ordered Dimensions is to multiplex an acyclic virtual network across the communication links of a cyclic network. Multiple Rings avoids deadlock, assuring that on each ring a node that has output a message must be prepared to accept a message from that ring. To allow rerouting from one ring to another, this system always maintains a free buffer for input for each ring when ring input is required. Finally, in the Tiny and Interval Labelling systems certain link-to-link connections are not allowed to avoid cyclic message routes.

Livelock freedom. In the adaptive strategy used in Tiny, livelock freedom is not implemented. It is also not implemented in the Ordered Dimensions system in which livelock freedom introduces a large overhead. On the contrary, in the other routing systems no livelock may arise. In oblivious sys-

tems livelock is avoided if the queue buffering is fair, though in adaptive systems this is not sufficient. In fact, to assure livelock freedom in Multiple Rings when the message fails to arrive at its destination within some interval, the routing relative to the rings, not Cartesian coordinates, is performed.

Adaptivity. Multiple Rings uses a strategy that adapts the message routing depending on the network load (the adaptive strategy of Tiny is not of much practical use). In contrast, CSN, Ordered Dimensions, Tiny/sequential, and Interval Labelling use oblivious routing. Thus, these systems are unable to avoid network congestion when the communication load in a network is unbalanced.

To show the relevance of this issue, look at the iPSC/2, which employs an oblivious routing system. The message latency increases almost linearly as the number of messages that simultaneously reach a common node.¹³ Adaptive systems, such as Multiple Rings, avoid this. Finally, note that although the Interval Labelling routing is oblivious, it should also implement the two-phase routing strategy that allows the elimination of network hot spots, although it adds some overhead.

Message transmission. Since the T800 generation of transputers does not implement direct link switching, messages must be entirely buffered at each intermediate node. This store-and-forward technique is implemented in Tiny, Multiple Rings, and Ordered Dimensions. CSN uses the more CPU-intensive virtual cut-through technique.

Interval Labelling uses wormhole routing. This method is very effective because the IMS C104 chips implement the routing in hardware. The message header traversing a network of IMS C104s creates a circuit through which the message flows. Thus a message can be passing through several IMS C104s at the same time without disturbing the processing transputers. This method minimizes the latency and separates communication from computation.

Network latency. Network latency is a major parameter to be evaluated in a routing system because high latency overhead may abate the benefits of parallelism in communication-intensive programs. Although on the basis of the measurements mentioned there is not a large difference among the internode latencies of the routing systems (the Interval Labelling cannot be compared), Ordered Dimensions outperforms the others. On the other hand, this system is not adaptive. This means that under a nonuniform traffic distribution the system performance will decrease dramatically, and the network may collapse.

Generally, the routing systems discussed here outperform the message-passing systems provided with the operating systems developed for transputer-based machines (Helios, Trollius). As mentioned before, the overhead of using a message-routing system hidden in an operating system is not small. For example, in Helios¹³ the read and write system library calls provide a communication system. Using these calls results in internode latencies for 64-byte and 256-byte

messages of 1,150 and 1,300 μ s, if the test nodes run only the processes performing the measurements (that is, no load).

When compared with the communication latency obtained for other multicomputers, such as the iPSC/2 and Ncube/10, the network latency of the message-routing systems presented here show their effectiveness. Two reports^{14,15} show the experimental internode latencies of the communication systems of the iPSC/2 and Ncube/10. With a message size of 64 bytes and very low demand, they are respectively 400 and 600 μ s. With 256-byte messages, the internode latency is 780 μ s for the iPSC/2 and 1,060 μ s for the Ncube/10. With the same message sizes the systems presented here offer similar or better performance.

This comparison shows that these routing systems for transputer-based multicomputers are more than comparable with the hardware-implemented message-passing system of the Intel iPSC/2, and much more effective than the software-implemented communication system of the Ncube/10.

Generality. Generality is an important feature to promote a large use of a message-routing system. All the message-routing systems presented here can be used on a great variety of hardware topologies. In particular, Tiny, CSN, and Interval Labelling base their routing on tables, so they can be used on any network topology that can be set up by the four transputer links. Multiple Rings can be applied to all of the computer networks where at least one Hamiltonian circuit can be identified (ring, mesh, torus, etc.). The Ordered Dimensions system can be used in any regular network topology.

ONE OF THE MAIN PROBLEMS that must be faced in achieving a broad use of parallel computers is the lack of tools that hide the physical architecture and offer a high-level interface to a user. The routing systems presented here implement a virtual level that makes machine topology irrelevant from the programmer's point of view.

These five routing systems for transputer networks include four in use in current implementations that allow data exchange between processes mapped on transputers not directly connected. They are efficient software implementations of routing algorithms that can be used on a great variety of hardware topologies.

The fifth system (Interval Labelling) will be the routing system of the latest generation T9000 transputer. If the simulation results are confirmed in the real use of the IMS C104 chip, the Interval Labelling routing system will represent a very good solution for message routing in the next generation of transputer-based multicomputers.

Next-generation parallel computers will be multiuser general-

purpose machines. To achieve this goal, efficient routing systems are needed. Therefore, message-routing systems will represent a critical aspect in the definition of a general-purpose parallel machine based on transputers or other processors.

On the other hand, the use of a general-purpose routing system allows the programmer to develop parallel programs without worrying about the details of the hardware configuration. This increases code portability and reusability. The approaches discussed here are in accordance with this technological trend that will bring many benefits to the implementation of real applications on multicomputer systems. ■

Acknowledgments

The CNR Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo (Program on Information Systems and Parallel Computing) partially supported this work under grant 90.0076.69. A preliminary version of this article appeared in the *Proceedings of the WoTUG 15th Technical Meeting* published in 1992 by IOS Press, Amsterdam.

References

1. L. Clarke and G. Wilson, "Tiny: An Efficient Routing Harness for the Inmos Transputer," *Concurrency: Practice and Experience*, Vol. 3, No. 3, John Wiley & Sons, New York, June 1991, pp. 221-245.
2. M. Cannataro et al., "Design, Implementation and Evaluation of a Deadlock-Free Routing Algorithm for Concurrent Computers," *Concurrency: Practice and Experience*, Vol. 4, No. 2, John Wiley & Sons, Apr. 1992, pp. 143-161.
3. D. May, "Transputers and Routers: Components for Concurrent Machines," *Proc. Third Transputer/Occam Int'l Conf.*, IOS Press, Japan, May 1990.
4. Meiko Scientific Limited, *CSTools User Guide*, Bristol, UK, 1990.
5. P.R. Miller, C.R. Jesshope, and J.T. Yantchev, "The Mad-Postman Network Chip," *Proc. Transputing*, IOS Press, Amsterdam, 1991, pp. 517-536.
6. M. Shumway, "Deadlock-Free Packet Networks," *Proc. Second North American Transputer Conf.*, IOS Press, Amsterdam, Oct. 1989, pp. 139-177.
7. J.T. Yantchev and C.R. Jesshope, "Adaptive Low-Latency Deadlock-Free Packet Routing for Network of Processors," *IEE Proc.*, Vol. 136, No. 3, IEE, 1989, pp. 178-186.
8. D.A. Nicole, E.K. Lloyd, and J.S. Ward, "Switching Networks for Transputer Links," *Proc. Ninth OUG Tech. Meeting*, IOS Press, Amsterdam, 1988, pp. 147-165.
9. W.J. Dally and C.L. Seitz, "Deadlock-Free Message Routing in Multiprocessor Interconnection Networks," *IEEE Trans. Computers*,

- Vol. C-36, May 1987, pp. 547-553.
10. A.W. Roscoe, "Routing Messages Through Networks: An Exercise in Deadlock Avoidance," *Parallel Programming of Transputer-Based Machines*, Proc. Seventh OUG Tech. Meeting, IOS Press, Amsterdam, 1988, pp. 55-79.
 11. A. Ciampolini, A. Corradi, and L. Leonardi, "Testing Parallel Objects Kernel Environment (POKE)," Tech. Report Progetto Finalizzato Sist. Inf. e Calcolo Parallelo, No. 3/75, C.N.R., July 1991 (in Italian).
 12. J. van Leeuwen and R.B. Tan, "Interval Routing," *The Computer J.*, Vol. 30, No. 4, 1988, pp. 298-307.
 13. O. Frieder et al., "Experimentation with Hypercube Database Engines," *IEEE Micro*, Vol. 12, No. 1, Feb. 1992, pp. 42-56.
 14. J. Powell, "Performance Issues," *Helios 1.2 User Manual*, Perihelion Software Ltd., Nov. 1990.
 15. X. Zhang, "System Effects of Interprocessor Communication Latency in Multicomputers," *IEEE Micro*, Vol. 11, No. 2, Apr. 1991, pp. 12-15, 52-55; correction June 1991, p. 6.

Additional readings

- Close, P., "The iPSC/2 Node Architecture," *Proc. Third Conf. Hypercube Concurrent Computers and Applications*, SIAM, Philadelphia, Penn., Jan. 1988.
- Debbage, M., M.B. Hill, and D.A. Nicole, "Virtual Channel Router Version 2.0 User Guide," Tech. Report, Dept. of Electronics and Computer Science, Univ. of Southampton, UK, June 1991.
- De Carlini, U., and U. Villano, "The Routing Problem in Transputer-Based Parallel Systems," *Microprocessors & Microsystems*, Vol. 15, No. 1, 1991, pp. 21-33.
- Felperin, S.A., et al., "Routing Techniques for Massively Parallel Communication," *Proc. IEEE*, Vol. 79, No. 4, 1991, pp. 488-503.
- Fielding, D.L., et al., "The Trollius Programming Environment for Multicomputers," *Proc. Third Conf. NATUG*, IOS Press, Amsterdam, 1990, pp. 207-210.
- Garnett, N.H., "HELIOS—An Operating System for the Transputer," *Parallel Programming of Transputer Based Machines*, Proc. Seventh OUG Tech. Meeting, IOS Press, 1988, pp. 411-419.
- Hu, L.R., and G.S. Stiles, "Fluid Dynamics on EXPRESS: An Evaluation of a Topology-Independent Parallel Programming Environment," *Proc. Fourth Conf. NATUG*, IOS Press, Oct. 1990.
- Inmos, *The T9000 Transputer Products Overview Manual*, SGS-Thomson Microsystems, 1991.
- Knowles, A., and T. Kantchev, "Message Passing in a Transputer System," *Microprocessors & Microsystems*, Vol. 13, No. 2, 1989, pp. 113-123.
- Pancake, C. M., "Software Support for Parallel Computing: Where Are We Headed," *Commun. ACM*, Vol. 34, No. 11, 1991, pp. 53-64.

Surridge, M.W., "ECCL: A General Communication Harness and Configuration Language," *Proc. Second Int'l Conf. Applications of Transputers*, IOS Press, 1990.

Syscon Corporation, "VCX User Guide," Columbia, Md., 1990.

Tanenbaum, A.S., *Computer Networks*, Prentice-Hall, Englewood Cliffs, N.J., 2nd ed., 1989.



Domenico Talia is currently a member of the CRAI research team in the CNR-PFI project. He works in the area of parallel and distributed systems and is a senior researcher fellow of the CRAI parallel computing team. His technical interests include distributed systems, parallel architectures, and concurrent programming languages.

Talia studied physics at the University of Calabria, Italy. He is a member of the IEEE Computer Society, the Association for Computing Machinery, and AICA (the Italian Association of Informatics and Automatic Computing).

Address any questions concerning this article to the author at CRAI, Localita S. Stefano, 87036 Rende, CS, Italy; dot@crai.it.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 165

Medium 166

High 167



Ware Myers

Contributing Editor

Get to market faster with FPGAs

Application-specific integrated circuits (ASICs) and mask-programmable gate arrays take time, often measured in months, to procure from a vendor. A field-programmable gate array (FPGA), in contrast, can be incorporated in a product within a few days after logic designers complete their work.

The time saved gets the product to market that much sooner. Moreover, if system test reveals a flaw, designers can correct it in another day or two. Later, if use of the product by early customers discloses a misunderstanding of the requirements, that, too, can be quickly corrected.

That fast implementation sounds promising. In fact, the use of FPGAs still lags behind gate arrays and cell-based ASICs, according to R.H. Krambeck, C.T. Chen, and R.Y. Taui of AT&T Bell Laboratories, writing in a Compcon 93 paper. "For most of the 1980s, ... highly complex designs for which a large unit volume was expected were designed as cell-based ASICs," they said. "Somewhat simpler or lower volume circuits were done as gate arrays." The reason, they continued, was that FPGAs were inferior to the other two approaches in three key areas: gate density, system clock speed, and ease of use.

The FPGA is still a new product, less than 10 years old. Altera Corporation introduced one of the earliest versions in 1984. Containing eight macrocells, "the EP300 had a programmable I/O architecture as well as a programmable logic array," S. Kopec reported to Compcon 90. "This structure allowed a single device to implement arbitrary mixes of registered and combinatorial logic in a single chip."

By the time Kopec spoke, "the complexity and capability of programmable logic devices had increased by over 20 times." FPGAs would soon be competitive for many applications with cell-based

ASICs and mask-programmable gate arrays.

Three years later Jesse Jenkins of Jenkins Research organized an all-day track at Compcon 93 to update progress in FPGAs. Papers from Altera Corp., Actel Corp., AT&T Bell Laboratories, Aptix Corp., Concurrent Logic Inc., Intel Corp., and VLSI Technology reported that gate-equivalent densities in the 5,000 to 10,000 range are now common. Several companies have exceeded 20,000. System clock rates range from 60 to 125 MHz. Clock-to-output delay is on the order of 10 ns. Pins run from 100 to 288.

With capabilities of this order, FPGAs can satisfy a large fraction of the fast-time-to-market applications. Craig Lytle estimates that Altera's new family, for instance, with densities up to 24,000 usable gates, is competitive in nearly 50 percent of gate-array design starts.

Some system operations have to be performed at a rate faster than a microprocessor, particularly a small, inexpensive one, can operate. For instance, "data capture is a hardware-intensive function," Jenkins observed. "It is difficult using software to synchronize controller operations with external data and balance memory requirements."

To match a microprocessor to a task of this sort Jenkins suggested:

- *Identifying tasks that must operate at full speed and those that can keep up at a slower speed.* For instance, an activity that must be completed in less than the cycle time of the processor has to be implemented in fast-response hardware.
- *Identifying operations that can be efficiently handled by microprocessor data paths and those that require high-speed logic in hardware.* An activity where speed is not critical is a candidate for handling by the micro-

processor with software.

- *Maximizing tasks that can be done by software and minimizing those that require hardware logic.*
- *Identifying asynchronous tasks that must operate with custom hardware because the pace required exceeds the interrupt capability of the processor.*

As an example, Jenkins outlined the design of a microcontroller accelerator for an instrumentation application. The design combines an inexpensive personal computer with an add-on hardware unit capable of capturing incoming data at a 40-MHz rate.

IEEE Membership— The Professional and Technical Edge

IEEE is the world's largest technical professional organization in electrical and electronics-related fields.

Our membership includes leading practitioners in theory, research and development, systems and equipment design, testing, and application, project, and technical management.

**Is your management
aware of the value
of your
IEEE membership?**



The Institute of Electrical and
Electronics Engineers, Inc.
445 Hoes Lane
Box 1331
Piscataway, NJ 08855-1331
(908) 562-3900

The potential Achilles heel in using large FPGAs for hardware add-ons, however, is ease of use. Converting logic equations, logic diagrams, netlists, schematics, hardware design languages, or other outputs of the logic-design process to the inputs needed to program the FPGA would be very time-consuming if it had to be done manually. To meet this need, all the companies have software tools, such as chip compilers, that provide largely automatic methods of performing this conversion. Some of these tools also interface with CAD tools supplied by outside vendors.

The output of these FPGA design tools is a data stream that programs the FPGA chip. Programming the chip means establishing the pattern of connections between the logic elements that configures the FPGA for its intended application. FPGA fabricators generally use two methods of making these connections: permanent and configurable.

Representative of the permanent connection is the antifuse element, employed by Actel, VLSI Technology, and others. This element is a dielectric material at the junction of two metallic lines. In its original state, it is an insulator and the connection is open. When programmed, by applying a high-voltage current to it, the material converts to a conducting state. This conversion is a one-time operation. It permanently programs the FPGA. If change becomes necessary, the designer would have to program a new FPGA.

Configurable FPGAs are programmed at power-up by downloading digital data to preset or clear flip-flops or other logic elements. In some applications the data stream is serial; in others, parallel. It may be obtained from a PROM or system RAM, or it may be downloaded from the system controller or microprocessor. One of Intel's FPGAs has a nonvolatile storage array on the FPGA chip itself that provides the data to configure the chip.

Configurable FPGAs are also reconfigurable by the simple process of loading a new set of configuration data. Reconfigurability has obvious advantages in adapting a chip to the needs of an application as they become apparent. It also makes it possible to operate a product in several modes, simply by reconfiguring its FPGAs between modes in a matter of milliseconds.

"With many different programmable logic solutions from which to choose, a design engineer is faced with an overwhelming task in effectively choosing the best programmable architecture to meet his needs," Jay Sturges of Intel said. The *Compcon 93 Digest of Papers* describes half a dozen of these possibilities.

Sturges, also in the same *Digest*, had a different objective. He outlined a quantitative approach to making a choice among the FPGAs available. He contends that there are many metrics, such as number of pins per module, ratio of fan-in to fan-out, and others, that help. Moreover, these metrics are mathematically related. For instance, when two are known, a third can be predicted. Consequently, a designer knowledgeable in these matters has a long leg up toward making a wise selection.

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 186 Medium 187 High 188



Richard H. Stern

Oblon, Spivak,
McClelland, Maier &
Neustadt, P.C.

1755 Jefferson Davis
Highway, Suite 400

Arlington, VA 22202

Glitches left in software copyright system

Maybe after the *Altai* and *Sega* cases you thought you no longer had any problems with copyright law interference when carrying on a career as a software professional. Think again. There are still plenty of features, glitches, or bugs (depending on how you see it) left in the software copyright system.

You will recall that in the *Sega* case the Ninth Circuit federal appeals court held that reverse engineering of software by disassembling code was permissible, if undertaken for "a legitimate purpose." In that case, the legitimate purpose was cracking the lock of a security system that kept "unauthorized" software out of the copyright owner's hardware platform. (The court did not explain what a legitimate purpose was for other cases. That will have to be determined by a possibly long and painful case-by-case process. The copyright statute does not address this issue, like most difficult points; therefore, judges must work out the answers in individual cases.)

In the *Altai* case, the Second Circuit held that copyright law had to be interpreted, to some extent, in the light of the needs of the computer software industry. That meant that software publishers' desires to protect "look and feel," "program structure," and "other nonliteral aspects" of software had to be scrutinized closely. This would occur when the effect of copyright protection would be to close off programmers' access to public domain features, functionally dictated aspects of programs, and features commanded by good software engineering practices. Also, programmers' "appropriation" of certain features from copyrighted computer programs would not be copyright infringement. So far, all to the good.

That stretch of software copyright law now seems more or less settled, for the time being.

At least it is settled in the Ninth Circuit (West Coast states) for reverse engineering and in the Second Circuit (New York) for functional features. It will remain settled unless and until the losers in those cases talk the US Congress into legislatively overturning those decisions, which an IEEE document says they are trying to do. But there are other weak spots in the software copyright dike—and the heroic little Dutch boy, whoever that is in our parable, has only so many thumbs with which to plug holes in the dike. The propagation rate of holes may exceed the propagation rate of thumbs.

But the Ninth Circuit—praised for its *Sega* decision—has just created what may seem to be a new software copyright glitch. Loading software into RAM creates an infringing copy of the software and is therefore copyright infringement. That is what the Ninth Circuit just ruled (Apr. 7, 1993) in *MAI Systems Corp. v. Peak Computer Inc.* MAI manufactures and sells computers, creates and markets operating system software for the computers, and sells maintenance service for the computers. Several MAI employees left MAI to form Peak, a repair and maintenance service company. The usual trade secret, copyright infringement, and unfair competition suit followed—of which the copyright infringement aspect seems most troublesome.

MAI licenses its customers to use its system software. When Peak services a computer belonging to one of MAI's customers, Peak turns the computer on. That boots up the computer, loading the operating system software (from a hard disk, floppy, or ROM) into the RAM of the computer. Absent permission from MAI, the court held, that action makes a copy of the computer program, and it is copyright infringement.

Peak sought to defend on the ground that an

**These are not
glitches or bugs in
the system. They
are features!**

infringing copy of a copyrighted work is made only when the alleged copy is "fixed" in a permanent or stable form, permitting it "to be perceived, reproduced, or otherwise communicated for a period of more than transitory duration." (This is a quote from section 101 of the Copyright Act.) Ephemeral copies, such as transitory images on a display screen, are not sufficiently "fixed" for them to be infringing copies under the copyright law. (The traditional copyright law analogy for this is that a poem written in the sand near the sea is not fixed in a copy, because it will be obliterated by the next tide.)

Peak argued, as probably most observers had believed thus far, that a copy in RAM is ephemeral, rather than fixed. It disappears when the computer is turned off, and the RAM is therefore no longer powered—the way the light from a light bulb disappears when you turn off the power. (At least, that is what happens in a PC. Perhaps, MAI's computers were left on all of the time—although that seems unlikely, unless they were mainframes.)

The Ninth Circuit disagreed, asserting that it is "generally accepted" that loading software into a computer creates a fixed copy of the software. The court was slightly troubled by the fact that the authorities it relied on for the alleged general acceptance did not reveal whether the software loading constituting the creation of an infringing copy was loading into RAM, a hard disk, or a ROM. No matter, the court said. A copy in a RAM can be "perceived, reproduced, or otherwise communicated." The court did not mention

for how long, but it felt satisfied that "the loading of MAI's operating system software into RAM, which occurs when an MAI system is turned on, constitutes a copyright violation." Therefore, the court found Peak guilty of copyright infringement and enjoined its conduct.

To be sure, section 117 of the Copyright Act permits the owner of a copy of a computer program to load it into the owner's computer. But section 117 did not apply here, the court held, because MAI's customers are licensees of the software, not owners of copies.

In addition, Peak apparently owned several MAI computers and offered to loan them to customers. (Peak ran advertisements offering loaners to potential customers, for use while their own computers were being repaired.) The court held that this conduct would infringe MAI's exclusive right to distribute its copyrighted software, and the court therefore enjoined Peak from offering loaners—end of case, or this part of it.

Arguments can be made that this is a sound or unsound interpretation of the copyright law. For example, perhaps owners of computers have a right to get them repaired, by whomever they please—making the conduct of the repair company, when it acts at the behest of the computer owner, a "fair use" of the operating system software. Perhaps, computer owners have an implied license to do this, or perhaps MAI is estopped from preventing them from doing so, once it has sold them the computer. Perhaps, the asserted difference between selling and licensing the software in this context is a sham, and computer owners' property rights govern over legal fancy footwork. For example, if the ROM BIOS of the computer is licensed, rather than sold to the purchaser of the computer, should that legal label be taken seriously?

On the other hand, this may well be a proper interpretation of the US copyright law precedents developed over the last 200 years for poems, novels, paintings, music, sculptures, maps, and

telephone directories—which govern here. These are not glitches or bugs in the system. They are features!

The point is that you can continue to expect interesting features of copyright law to turn up in software copyright cases, so long as copyright law is the statutory mechanism that we use to regulate software conduct. It would be unreasonable in the extreme to expect copyright law to be turned inside out, just to accommodate software professionals. Copyright law has a long and venerable history; it has its own traditions. To alter them just to suit software creators and users would run the risk of destroying traditional rights and benefits enjoyed by beneficiaries of traditional copyright law—publishers, artists, composers, motion picture studios, playwrights, poets, novelists, whatever. Indeed, to do that you might have to change copyright law so much that it would no longer be copyright law. (That is what the book publishers decided when the chip industry tried to get Congress to pass a chip layout copyright law. They kicked chip layouts completely out of the copyright law and put them under the *sui generis* Semiconductor Chip Protection Act of 1984, instead.)

Here's a different kind of glitch, although this one really deserves to be recognized as a deliberate feature of copyright law. The Eighth Circuit (North Central states) has just ruled (Apr. 6, 1993) in *National Car Rental System Inc. v. Computer Associates International Inc.*, that state law rather than federal copyright law governs the extent to which a software licensee may use a licensed computer program. CA licensed NCRS to use CA's computer programs "only for internal operations and for processing its own data." NCRS later started using the computer programs to process data of its truck leasing and car rental subsidiaries or affiliates. CA found out and threatened to sue.

NCRS then brought a declaratory judgment action, admitting the use, but

contending that such use was within the scope of the license and was not copyright infringement. CA then countersued, asserting breach of the license contract (a claim based on state law) and copyright infringement (a claim based on violation of federal law). NCRS riposted by asking the court to dismiss the breach of contract claim that CA brought under state law as being preempted by the federal copyright law involved in CA's second claim.

Section 301 of the federal copyright law provides that federal copyright law preempts state law claims when they are equivalent to copyright law claims. That means that state law cannot regulate the same thing that federal copyright law regulates. A state cannot authorize or prohibit copyright infringement, for example, because that would interfere with copyright law's uniform regulation of the field.

When is a state law equivalent to copyright law? That is the central problem in copyright preemption cases. Typically, the case turns on whether the state law claim has one or more extra elements in it (such as use of force, carrying away of property) that are not necessarily found in a federal copyright infringement claim, and are qualitatively different from the elements of the federal copyright claim.

Here, the trial court held that CA's copyright infringement claim was the same as its breach of contract claim, and it therefore dismissed the latter. The trial court said that both of CA's claims were directed to the same thing, unauthorized utilization of the copyrighted computer program, in this case amounting to a computer program lease going from NCRS to its affiliates.

The Eighth Circuit found the controversy to raise a close point on equivalency. But it concluded that the provision in the license contract between CA and NCRS limiting how NCRS was to use the computer program constituted an extra element not found in a copyright infringement claim. The Court thought this would be true even though a use of intellectual

property in excess of the scope of a license is infringement. Here, the contract goes beyond CA's rights under the copyright laws. It requires NCRS not to use the software for the benefit of third parties. Accordingly, the Eighth Circuit reversed, holding that the contract claim under state law should be tried along with the federal copyright infringement claim.

Probably, the Eighth Circuit read copyright law properly. Congress did not intend to make copyright law cut off contract claims, in the ordinary case. Sometimes, contract claims and copyright infringement claims are equivalent. But these claims are probably not equivalent, because of a qualitatively different extra element.

What is the result, however? Consider a software company that has a nationwide marketing policy and a standard contract that it uses throughout the United States. Whether restrictions in it, such as the use restriction involved in this case, are valid and enforceable depends on the laws of 50 different states. Perhaps that is a rational choice for the ordinary subject matter of copyright law. But it is not a sensible way to regulate a nationwide software business.

It would make more sense if customers' violations of restrictions in software licenses were judged under a federal standard. They should be held copyright infringement if the vendors' restrictions are legitimate from a standpoint of software policy. They should be held permissible conduct (not violations of state or federal law) if the vendors' restrictions go too far from the standpoint of a federal law's software policy.

That is not the present law, of course. That could be the law only if Congress passed a uniform software regulatory law for the whole country. One provision of it would be a rule on when restrictions on customers' use of computer programs were permissible and thus customers' disobedience of the restrictions were infringement. That would provide a uniform federal system regulating software conduct. It would provide known, predictable

**We need a system
whose features
don't look like
bugs.**

remedies for violation of the regulatory standard with resulting certainty and security of expectations for software users and software marketers.

There are many pros and cons of having such a regime, and the arguments generate considerable heat. (A professorial representative of one important mainframe company has just published a lengthy article heatedly denouncing *Sega*, *Altai*, and the "false icon of *sui generis* protection" for software. He argues that it would be a great mistake to adopt a "genre-specific *sui generis* regime invented out of whole cloth" as an alternative to the "growing experience under the present copyright law and the increasing certainty that it provides" for software as "court decisions by degree crystallize into an understandable and sensible doctrinal matrix.")

From time to time in the coming months, I will address various aspects of such issues and comment on different observers' positions. Much later this year, I will report on a conference at a major national school of engineering. This conference will be devoted to the issue of whether it is time to explore alternative models of software protection law. Such alternative models would differ from the pure patent law or copyright law models (as I prefer to term them) or paradigms (as many of my nonengineer professor friends prefer to term them). Perhaps the time has come to consider reinventing the form of intellectual property or industrial property protection that we use for software, or for some kinds of software.

Perhaps that could lead to a system whose features did not look like bugs.

Guest review

Of general interest to readers is the following book review by John F. Noble, Washington, D.C., attorney and editor in chief of the *Computer Law Reporter*, a monthly journal of computer law and practices. Noble's full comments on the current battle for control of the software industry appeared in the April 1993 issue of *CLR*; what follows is a shortened version.

Softwars: The Legal Battle for Control of the Global Software Industry, Anthony L. Clapes (Quorum Books, Westport, Conn., 325 pp.)

A war rages over the extent to which the original developers of computer software should be protected from the sincerest form of flattery. Both sides of the battle will enjoy Clapes' account of this high-stakes war. Unencumbered by a lot of technical and legal jargon, Clapes mixes a litigator's war stories with a cogent philosophy of what the law should be. He adds just enough economic analysis to anchor the debate in the real world.

Clapes is not an unbiased correspondent. As IBM's assistant general counsel in charge of litigation, he is one of the warriors, and his allegiances are obviously and admittedly with the "innovators" in their campaign against the "imitators." This does not detract from the book. Indeed, an author pretending to even-handedness could not have portrayed the fierce pitch of the battle that is conveyed by Clapes' sometimes acerbic account. At the risk of pushing the metaphor, you can almost smell the blood.

Softwars recounts the development of the law regarding the application of patent, trade secret, and particularly copyright protection to computer software, from *Apple v. Franklin* in 1982 to *Nintendo v. Atari* in 1992. In the course of this exposition, the author takes the reader to courtroom battlefields in Philadelphia and Northern California, across the ocean to Japan and Australia, and into cyberspace

where a would-be astronomer tracks down a pirate named Hunter in the employ of the KGB.

The theme of the book is that "the outcome of the legal battles over software will determine the nature of the industry for the foreseeable future, and the nature of the industry will dictate the identity of the firms that will be most successful in competing in that industry." As its author says, "The prize is the computer industry itself."

This war, according to Clapes, will determine whether the computer industry in the future is marked by innovative competition between firms that rely on the development of advanced products, or imitative competition between firms that compete primarily on price.

Clapes weaves legal and economic arguments in support of the proposition that the vitality of the industry depends upon providing legal protection to the creative work of programmers and the capital investment of their sponsors.

The legal argument proceeds from the premise that the creative nature of the work, and Congress' explicit extension of copyright protection to computer programs, entitles software to protection under the traditional principles of copyright law. Under those principles protection extends to the nonliteral elements of original expression, and applies in particular to the program's "interface" with the user.

The seminal case for this "traditional" application of copyright law principles to software was the 1987 decision in *Whelan Associates, Inc. v. Jaslow Dental Laboratories, Inc.* *Whelan* held that an accounting program for dental laboratories was infringed when a competing program adopted its "sequence, structure and organization." In that case, the Third Circuit Court of Appeals applied the principle that only expression and not the underlying idea of a work is protected by copyright. It held that the unprotectable idea of a utili-

tarian work like a program is its purpose or function, and that the protectable expression was everything that was not necessary to the purpose or function.

Clapes acknowledges that the case was "highly controversial." But he insists that the decision represented "nothing more than the application of traditional copyright principles to a case involving computer programs." He is not the least embarrassed to defend the decision in *Whelan*. Still, one might wonder about his characterization of the decision as a "victory, *some conclude* for the purveyors of proprietary software." Would he agree that it has turned out to be something of a Trojan horse? In the subsequent case law and commentary, its analysis of the idea/expression dichotomy has almost unrelievedly been described as simplistic and overbroad.

For a time, it seems, the "innovators" were on the offensive. Clapes applauds Judge Keeton's decision in *Lotus v. Paperback*, decided in 1990. Keeton held that the copyright law does not treat computer programs differently. Therefore, nonliteral elements of the 1-2-3 spreadsheet program, such as its menu structure, were entitled to protection under traditional copyright principles.

Clapes recounts the reaction to Keeton's "lodestar" decision as the opposition regrouped. The so-called "gang of ten" copyright law professors rose up to advise the Court that its copyright analysis was contrary to the statute, case law, and traditional principles of copyright. The League for Programming Freedom picketed Lotus' headquarters. Law firms with clientele in the enemy camp "salted" the legal literature with articles critical of the opinion in *Lotus*. One commentator, according to Clapes, claimed that the judge had given Lotus the exclusive right to "use the F1 key for a Help function."

Forward in time, and across the country, Clapes turns his attention to *Apple v.*

Microsoft. In that case, Judge Walker rejected the claim that Microsoft's Windows appropriated the overall look and feel of the Macintosh user interface, instead examining isolated elements of the Mac interface for protectability.

Some might conclude that the evil empire of imitators was resurgent. But Clapes offers a more sanguine analysis. He points to an earlier license agreement between Apple and Microsoft. Clapes argues that "just as a separation agreement complicates a couple's lives, the license agreement complicated Apple's prosecution of its case."

As a result, according to Clapes, the judge concluded that he was required to separate the licensed elements of the interface before comparing the two works. According to Clapes, "[N]ormally, a copyright plaintiff, particularly one who is asserting copyright in graphical images, is entitled to have an infringing work evaluated against the original work on the basis of the *gestalt* of the two works: the overall impression or 'total concept and feel' that they convey."

There is no particular citation to support this broad statement of the law, and the issue is a closer one than Clapes will allow. One can almost hear the "gang of ten" collectively muttering, "in your dreams, pal."

Clapes would reserve a special place in hell for the advocates of "reverse engineering." This defense to appropriation of intellectual property found favor with the Supreme Court in *Bonito Boats, Inc. v. Thundercraft Boats, Inc.* The *Bonito Boats* decision stated that the reproduction of a fishing boat hull design, claimed as a trade secret, by making a mold of the original was not unlawful. It was legal because "[t]he public at large remains free to discover and exploit the trade secret through reverse engineering of products in the public domain."

The reverse-engineering defense has been invoked in the software context as a defense to the practice of discovering the programmer's trade secrets by run-

ning a program's object code through a reverse compiler to replicate, or at least approximate, the original source code. This, complains Clapes, allows a competitor to "unlock the secrets of an original program" by "peeking."

Clapes argues strenuously that reverse compiling is different from the kind of reverse engineering approved by the Court in *Bonito Boats*. The shape of a hull enters the public domain when it leaves the factory. Unlike this, a computer program, on the other hand, keeps its secrets by being marketed in machine-readable object code, and sold subject to license agreements that prohibit reverse compilation. According to Clapes, "Reverse engineering" is a term that makes no real sense when applied to software. The use of the term is a form of propaganda that obscures what is really going on."

In Clapes' view, reverse compiling a program is indistinguishable from translating a French novel to English—a right reserved to the author.

He undermines his own argument, however, by attempting to rebut the antiprotectionist argument that reverse compilation is necessary and appropriate to the understanding of the program. Understanding is necessary to the dissemination of its ideas, which is in turn one of the bedrock purposes of copyright law. Although he takes issue with this articulation of the purpose of copyright law, his fallback position is that object code is not unreadable at all.

If Clapes is right that "[s]ome programmers can read object code[,] and that [m]any more can decipher it with effort," it would seem that he is in the same boat with *Bonito*. His secret is not so secret. The legitimacy of the endeavor to uncover the "secret" surely does not turn on the arduousness of the task.

Clapes tours battlefields in Australia and Japan where the reverse engineers were routed by AutoCAD and Microsoft. His description of the Japanese legal system is informed and fascinating. His account of the Australian case about a hacker who bypassed

**"The prize," says
the author, is the
computer
industry itself."**

AutoCAD's security system proves that Clapes is not totally shameless in his protectionist sensibilities. The hacker was found guilty of infringing a copyright in an electrical circuit.

The Gettysburg of the Softwars is *Computer Associates v. Altai*, a case that has assumed a significance far surpassing the initial stakes.

The case concerned an interface program called Adapter, developed by Computer Associates and copied by an Altai employee.

The case would have been unremarkable but for Altai's response to CA's copyright infringement and trade secret misappropriation complaint. Altai rewrote the program to eliminate the portions copied verbatim from CA. Oscar 3.5 was the result.

CA amended its complaint to allege that Oscar 3.5 was also an infringement of its copyright and a misappropriation of its trade secrets. Because there was no direct evidence of copying, the Court was required to determine whether Oscar 3.5 remained "substantially similar" to Adapter.

Judge George C. Pratt, a member of the Second Circuit Court of Appeals on temporary assignment in district court was the trial judge. He began what Clapes refers to as his "revisionist analysis" of the substantial similarity issue by observing that "in the context of computer programs, many of the familiar tests for similarity prove to be inadequate, for they were developed historically in the context of artistic and literary, rather than utilitarian works."

Clapes' understated characterization

of this premise as "questionable" does not totally convey the dismay it must have engendered in the protectionist camp. And it got worse. As recounted by Clapes, "[f]rom his shaky premise, Judge Pratt leapt to the even more tenuous conclusion that the *Whelan* case, without question one of the most important software copyright cases decided to date, was 'inadequate,' 'inaccurate,' 'simplistic,' and 'fundamentally flawed.'"

Leaving *Whelan* like an overturned tank by the side of the road, Pratt formulated what has become known as the abstractions-filtration-comparison test. That test is based on Judge Learned Hand's 1930 opinion for the Second Circuit in *Nichols v. Universal Pictures Corp.* It recognizes in a work "patterns of increasing generality" as "more and more of the incident is left out," leading up to "the most general statement of what the [work] is about." Under this test, the court is called upon to identify a level of abstraction that will be the demarcation between protected expression and unprotected idea.

When Pratt applied his abstractions test to the Adapter program, what remained of Adapter in Oscar 3.5. Its parameter lists, macros, and "high-level architecture" fell, for the most part, on the unprotectable "idea" side of the dichotomy.

Clapes labors to minimize the significance of the decision in *Computer Associates*. He claims to be heartened by the fact that the Court, by citing *Nichols*, reached back to the "well-spring of presoftware precedents dealing with the idea/expression dichotomy." Clapes questions whether there are really "essential differences" between the two tests.

It may seem that Clapes is whistling past the graveyard here. After all, under the first test "structure, sequence and organization" is protected. Under the second, parameters, macros, and high-level architecture are not.

But I suspect that Clapes is more devious than oblivious. He is arguing

his case, in a well-crafted brief, before the judges and their clerks in nine other circuits, and one higher venue, which have yet to squarely address the issue.

The choice that Clapes would like to present is between *Whelan* and *Nichols*, not *Whelan* and *Computer Associates*. He is right that *Nichols* and *Whelan* are not that far apart, and the protectionists would be delighted to have the battlelines drawn there. The problem with this is that Pratt's decision merely uses *Nichols* as a legitimate point of departure, and he is a far piece down the road before he is finished. The antiprotectionists have advanced, and a new battleline is drawn between *Whelan* and *Computer Associates*.

As much as he might wish it were not so, Clapes is aware of this too. He takes some shots at Pratt's opinion, and the factors that produced it, in an effort to undermine its authority.

In particular, Clapes suggests that the outcome was unduly influenced by the technical expert—Randall Davis of MIT's Artificial Intelligence Laboratory—appointed by Pratt to help him interpret the technical evidence. Clapes credits the expert as being an "intelligent, articulate, and interested computer scientist ... known as a moderate" in the software protection debate. However, Clapes maintains that "everyone has a bias, and Davis is no exception." Davis' bias, in Clapes estimation, is that of a skeptic, inclined, in Davis' own words, to the view that the "old ways of doing business and the old ways of thinking may simply not work any more."

Clapes argues that Davis, in his unusual but not unprecedented role as an independent expert appointed by the Court, swayed the outcome of the case. He suggests that Davis' opinions were given greater weight because of his ostensible neutrality. Also, his bias toward "questioning the fundamental premises of intellectual property law" was not fully revealed at trial because the attorneys felt constrained to "treat him fairly gingerly."

Clapes also suggests that Pratt's "revisionist" analysis was born of an incli-

nation to seize the opportunity to assert the Second Circuit's reputation as the strongest copyright law circuit. He wonders: "Was the whole point of the rejection of the *Whelan* analysis to promulgate a Second Circuit test for nonliteral infringement instead of a Third Circuit test, a kind of Not-Invented-Here reaction? Perhaps." Clapes points out that Pratt's brethren on the Second Circuit, "not surprisingly," affirmed his decision.

Unfortunately, Clapes' account of what may prove to be the 100 Years Softwar apparently bumped up against his publisher's deadline. The Federal appellate decisions in *Atari v. Nintendo* and the Ninth Circuit in *Sega v. Accolade* receive only cursory treatment in a footnote.

If one makes appropriate allowance for the partisanship of its author, *Softwars* is a valuable survey of the battlefield's terrain. Newcomers to the field of intellectual property law will find it accessible. For lawyers already steeped in intellectual property law, the book provides a thoughtful and provocative perspective.

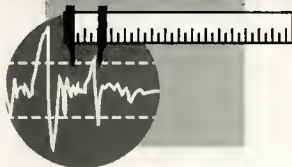
If the book has a failing, it is that Clapes' orientation does sometimes color his depiction of reality. At one point, for example, he writes: "There is no real uncertainty about copyright protection for software. Don't let anyone tell you differently." It is true, of course, that software is entitled to copyright protection. But this would not have been nearly so interesting a book if there were not a great deal of uncertainty about the *extent* of copyright protection that software is entitled to, and the *nature* of the conduct that will be deemed infringing.

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 177 Medium 178 High 179

Micro Standards



Organizing the corporate standards function

In today's world of "rightsizing" (a euphemism for downsizing), "downsizing" (a euphemism for layoffs), layoffs (a euphemism for not getting a salary), "leftsizing" (watch this space for details), and every other kind of "sizing" but "upsizing," standards participants are increasingly being called upon to justify their existence with short-term economic benefits. At some particularly shortsighted companies engaging in this sport du jour, a strategic rationale for standards without a "this-quarter-bottom-line-dollars" benefit is just short of useless.

While this situation is undoubtedly painful for some standards participants, I believe that it represents an opportunity to refocus a company's standardization efforts on their real function—helping to make the company's products successful. Such a focus on product success emphatically does not mean attempting to subvert the standards process for parochial ends. On the contrary, it means a commitment to working with standards processes over the long term, as an integral part of the development of successful products.

An analogy that I find useful is product quality. Attempting to achieve quality by bolting it onto an existing product is absurd—quality must be an integral part of the entire process of the company. The same is true of standards: decisions regarding participation in the development of standards, conformance to standards, or the creation of standards cannot successfully be treated as afterthoughts. To do so encourages terrible standards and unsuccessful products. Standards, like quality and the spices in spaghetti sauce, must be in there from the beginning.

There are many ways to organize standards within a larger company; some are very productive and others are not. It's very difficult, for ex-

ample, to internalize the "in there" approach where a centralized standards organization calls all the shots on standardization issues. In such a situation, the product groups—the engineering and marketing departments—either abrogate their product responsibilities and let the standards department dictate standards compliance and participation, or they ignore the standards organization entirely. Both outcomes are potentially disastrous while being wholly avoidable.

The key is matching the standards organization to the developmental stage of the corporation. Paradoxically, smaller companies often do a better job in standards than larger companies, at least in the short term. Because they usually can't afford to hire standards specialists, the engineering and marketing departments must be involved directly. Unfortunately, these companies often do not have the resources to make the necessary long-term standards investments.

Properly managing standards involvement in larger companies represents a "win-win" opportunity for both the company and the industry within which it operates. In this column I propose a three-stage internal standards organization developmental taxonomy (see Figure 1).

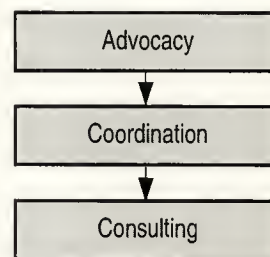


Figure 1. Internal standards organizational taxonomy.

Stephen L. Diamond

SunSoft, Inc.

Phone (415) 336-4190

Fax (415) 336-4477

steve.diamond@eng.sun.com

Standards advocacy

Larger companies may reach a significant level of standards involvement before they consider creating a standards department. Sometimes hundreds of engineers are participating in standards before a company recognizes the need for some focus. The good news is that people in the product groups are involved with standards because their products depend on them. On the other hand, such involvement in standards lacks any long-term commitment or focus and has little or no coordination. Also, little thought is usually paid to effectiveness or efficiency.

At this stage, the standards function should begin with a small team focused on advocacy. By "advocacy" I don't mean preaching standards as a panacea, rather the team should educate the product groups about the universe of standards development activities. This education should be quite focused, and cover standardization status, process, trends, and potential rewards. Product development groups should be aware of which standards organizations and activities impact their current and future products. They should understand how standards are developed, and how they can become involved. Finally, they should have a picture of future trends so that they are not blindsided when a standard is being developed or approved in their sphere of interest.

The standards team in the first stage might consist of four people: an engineer to serve as a liaison to the engineering community; a marketer to serve as a liaison to the marketing community; a standards expert; and a manager to bring the standardization message to company management. To avoid expectations that the standards team is there to "do standards," it is important not to build a large group. The product groups should "do standards;" the standards group is there to navigate—not to drive. If it becomes necessary to participate in a standards developing organization (SDO) creating a standard that the company wishes

to implement, the resources should come from the product groups, not a standards group. The goal of stage one is for the product development groups to achieve the "minimum adult requirement" of understanding of the standardization process, and to begin to address standardization requirements in all their product definitions.

"Address standardization requirements" doesn't mean that every prod-

***Every product
definition has to
consider
standardization,
both formal and
informal.***

uct must conform to a standard. It means only that every product definition has to consider standardization, both formal and informal. It may mean that the product absolutely does not conform to an existing standard, but it may also mean that the product team will have to participate in the development of a future standard. To address standards, the team must understand them. Team members must be able to assess the costs and benefits of the standardization equivalent to the make/buy decision: the lead/follow/ignore decision.

Where a standard exists, the product team has to choose whether or not to conform. Even if they choose to conform, team members still must decide whether to participate in the evolution of the standard in question, or simply to sit back and observe. Where no standard exists, the product team has to

decide whether or not to lead in the creation of a new standard. Another possibility is to ignore formal standards and attempt to create an informal, de facto standard, but the issues to be addressed are very similar. My point is that the product team must explicitly consider these decisions.

Standards coordination

Once the suggested minimalist standards infrastructure is in place, standardization activities ideally will begin to diffuse throughout the company. At this stage, the standards function should move from an advocacy role to a coordination role, with both an internal and external perspective. Internally, the now-standards-literate product development people will increasingly begin to make decisions regarding standardization. These decisions should make sense across the company; externally, with the plethora of standards developing organizations today, it is essential that a company's standards efforts are coordinated across organizations. For example, consortia, user groups, and SDOs overlap significantly in scope; one organization may attempt to develop a standard that overlaps with one developed elsewhere.

Standards consulting

Significant, distributed standards awareness and competence within the product development groups characterize the third stage. The standards function should metamorphose at this point into a proactive high-level consulting organization. Having liaisons into the engineering and marketing communities may no longer be necessary, because these communities will have already embraced the message and their need will be for specific information and consulting. Instead, standards specialists should replace these liaisons in the standards department. Maintaining a high level of management awareness of standards, however, is still important, preferably by a high-level management reporting relation-

ship. The ultimate function of a standards department should be to

- educate management, engineering, and marketing;
- provide information beyond the company's usual sphere;
- monitor and project standards trends;
- guide internal standards participants;
- consult; and
- help coordinate broad, long-range standardization strategies.

Standards people typically should not hold technical positions in standards bodies, I suggest, but rather should arrange that appropriate representatives from a product group be made responsible for standards by their management. On the other hand, a key function of the standards professionals at this stage is to participate in high-level policy-making positions within standards bodies. There are several reasons for this commitment. First, I believe that beneficiaries of the external standardization infrastructure have an obligation to work at improving it for the benefit of the industry. Second, such positions provide greater visibility and access and indirectly help the company. Finally, by virtue of their role within the company, the standards professionals are in the best position to encourage a fair and open process.

The meaning of time

A key factor that must be considered in the development of an internal standards organization, processes, and strategies, is time—not in the tactical sense but in the historical sense. Most standards organizations, especially SDOs, have evolved over years. The IEEE, for example, has been involved in standards development for over 100 years. Any company that expects to participate meaningfully in standards should treat that involvement as a long-term commitment, both to standards and to the industry. Companies that are

willing to make such a long-term investment will help themselves and their industry. Companies that seek to achieve short-term, parochial advantage in a long-term standards world will benefit neither.

Obvious benefits

In conclusion, let me raise the question of why standards participants are so often asked to justify their existence. Aren't the benefits of standards obvious even to management? The short answer is no. The long answer is that standards professionals, thinking that the benefits are obvious, often fail to adequately communicate them to their company's management. Another problem is that standards professionals frequently are reluctant to involve themselves in what they perceive as petty "commercial" issues, preferring to deal in the rarefied realm of standards organizations and their often hermetic and arcane policy, procedures, processes, and people. The solution is simple: by bringing standards into the product development process as an important and integral participant, the benefits of standards become obvious to everyone.

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 180 Medium 181 High 182

NEW TITLE

Computer-Aided Software Engineering (CASE) 2nd Edition

edited by Elliot Chikofsky

This new edition of the popular technology series on CASE describes new information on its technology, its background, and its evolution. The papers presented in its text illustrate the present state of CASE, how its concepts have fared over time, and how it looks as a technology for the future.

The second edition features more than 35% new papers and combines the latest key papers in the field with background articles that allow the reader to see how the field is evolving. It is also updated with current material on:

- * Integrated Environments
- * Tools and Assessment Evaluation
- * Process-Based Integration
- * Learning Curve
- * CASE Adoption Pitfalls
- * I-CASE

Sections: CASE Environments and Tools; Overview, Evolution of Software Development Environment Concepts, Role of Data Browsing Technology in CASE, Role of Assistants and Expert System Technology in CASE, Role of Prototyping in CASE, Tailoring Environments, Issues of Evaluating Tools and Managing CASE.

184 pages. January 1993. Softcover.

ISBN 0-8186-3590-8.

Catalog # 3590-05 — \$35.00 Members \$25.00

Order toll-free

1-800-CS-BOOKS

or Fax (714) 821-4010

(In California call (714) 821-8380)



**IEEE COMPUTER
SOCIETY PRESS**

Micro Review

Richard Mateosian

2919 Forest Avenue

Berkeley, CA

94705-1310

(510) 540-7745

Manuals and guest reviews

Nowadays it is entirely feasible for a small company or product development group to produce its own highly professional hardware or software manuals. One tool stands out as the best for this kind of job.

FrameMaker 3.0 (Frame Technology Corp., San Jose, Calif.)

I've used Microsoft Word for the Macintosh for many years, and recently I've become very fond of Word for Windows on PC systems (See Micro Review, Feb. 1993). For many jobs I wouldn't use anything else. For large jobs, however, the nod has to go to FrameMaker.

Unlike Word, which started on small systems and added features over time as those small systems became more powerful, FrameMaker began as a publishing system on workstations. As personal computers have evolved into sufficiently powerful systems, FrameMaker has migrated unchanged and intact to them. While preparing for this review, I worked with both Macintosh and Windows versions; they are virtually identical. Files are interchangeable between them. Since large jobs often require the collaboration of several writers using a variety of computers, this is an important feature. By contrast, Word for the Macintosh and Word for Windows have separate histories, but have evolved into similar programs. They are far from identical.

Another interesting difference between Word and FrameMaker arose when I tried to use each to read the other's files. FrameMaker easily opened a Word document and displayed it with all its fonts and formatting. Word, on the other hand, opened a FrameMaker document, and it looked like pages of gibberish.

In my review of Word 5.1 for the Macintosh I complained about how difficult it was to use on

the tiny screen of a Mac SE/30. This problem is far worse for FrameMaker. You can use it on a small screen, but it's practically impossible to work effectively there. In fact, it's difficult to use on the larger 640x480-pixel VGA display of my PC. The other side of this coin is that FrameMaker can make good use of, and justify the purchase of, the high-end machines of the PC and Macintosh lines.

FrameMaker allows you to make a book file that defines a book as a sequence of separate files. You can design page layout and paragraph and text formats for the book then use FrameMaker's capabilities to enforce uniform formats and continuous numbering of pages, figures, tables, and other sequences. You can build cross-references, an index, and a table of contents, and FrameMaker will update these for you whenever you ask it to. These tasks are easy to accomplish with FrameMaker, once you master its arcane ways of specifying them. They are almost impossible to accomplish with Word.

FrameMaker has a built-in drawing facility, and it can import graphics in a large variety of formats, either by actual inclusion or by reference. Importation by reference uses paths relative to the directory that contains the book file, so it is easy to keep the files together and move the entire package from one system to another.

Since FrameMaker maintains a uniform environment across platforms, its Macintosh and Windows versions don't look like standard Macintosh or Windows programs. This can be disconcerting. On the other hand, there are platform differences in areas like keyboard shortcuts. These can make switching between platforms a little rougher, but it's still pretty smooth.

The biggest lack I find in FrameMaker is a good macro facility. By comparison with

Microsoft's Word Basic, FrameMaker's capabilities are virtually nonexistent.

The other complaint I have about FrameMaker is that I often can't predict what it will do in response to what I think are reasonable commands. The worst examples I've seen of this involve trying to place and group graphics in anchored frames. Pieces of them sometimes wind up scattered all over the printed document. Either I don't understand how it works, or it has a few bugs. The whole program is so complicated that I'm not sure which is true. The documentation, while voluminous, has many gaps when it comes to details.

If you want to work across platforms or put together large documents, FrameMaker is the first product you should consider. It may well be worth the extra money it will cost you.

Compression

In the interest of disclosing any possible conflict of interest, I need to tell you that AddStor, Inc., a major supplier of compression products for PCs, is one of my technical writing clients.

Compression is the next big thing. It's an idea whose time has come. There is never enough disk storage and never enough transmission bandwidth. Every disk soon becomes too small, as programs and data expand to fill the available space. Every modem and every network soon becomes too slow, as larger programs and more data move from one place to another.

The first attack on this problem came from the proliferation of archiving programs like Stuffit and PKZIP. These simply take a collection of files and build a large file containing all of them. By encoding redundancy and eliminating the dead space at the ends of the individual files, the archiving program builds a combined file that takes up a lot less room than the sum of the individual file sizes.

As the popular personal computers became more powerful, they made possible another kind of compression. This is exemplified by PC programs like

Stacker and SuperStor, which perform compression and decompression in real time. These programs compress virtually all files on a disk into a single large file and install a driver program to allow accesses to that file as if it were a disk drive. The driver compresses as it stores and decompresses as it reads, and it all happens so fast on processors like a 486 that you never really notice the delay.

The main problem with programs like those described in the last paragraph is that you need to add them onto the operating system through the CONFIG.SYS and AUTOEXEC.BAT files on the PC or through similar mechanisms on other systems. The obvious next step was to include compression as an integral part of the operating system, and Microsoft has just done that.

MS-DOS 6 (Microsoft Corporation, Redmond, Wash., \$129.95)

MS-DOS 6 is the next major version beyond MS-DOS 5. Its main achievement is to incorporate features that large numbers of DOS users were adding to their systems from third-party sources. It includes virus protection and backup utilities that run from Windows, a memory manager, several levels of file-deletion reversibility, and compression.

The compression system included with DOS 6 is called DoubleSpace. You simply invoke the installation program from the DOS prompt, wait a few minutes for it to compress your files, and with no apparent change to anything else, you now have about twice as much disk space as you had before. I did this about a month ago, and I have had absolutely no trouble running all of my old DOS and Windows applications.

Microsoft claims that DOS 6 is more tightly integrated with Windows than DOS 5 was. I haven't noticed any difference along those lines. I certainly haven't had any trouble running Windows.

I can't think of any reason why you shouldn't run right out and upgrade to DOS 6.

Books

High-Speed Digital Design—A Handbook of Black Magic, Howard W. Johnson and Martin Graham (Prentice Hall, Englewood Cliffs, N.J., 1993, 458 pp.; \$45.95)

When I received this book a couple of weeks ago, I didn't look at it very carefully or really notice it. Then I ran into Marty Graham at a conference and saw him waving it around proudly, and I finally realized what I had received. After 40 years in this business, this is the first book U.C. Berkeley professor emeritus Graham has ever put his name on. If you are interested in high-speed digital design, you should drop everything and read it.

I don't know how the authors divided the creation and organization of the topics. Graham told me that Johnson, a specialist in high-speed digital communications and digital signal processing, did all of the writing. He also told me that every example came from the direct experience of one or the other of them. In other words, this is not a survey. It documents the experience of two respected specialists in a neglected field.

This book aims to alleviate a problem in the education of digital designers. Over the last couple of decades, the analog circuit principles that apply to high-speed digital design have fallen out of standard college curricula—for the simple reason that they are largely irrelevant at the speeds most designers have been working with. Now, however, as speeds increase, designers lack the training to deal with the "black magic" of managing high-speed effects.

You can learn the basic terminology, the high-speed properties of logic gates, and the standard measurement techniques by reading the first 130 pages. After that you can skip around among the specialized topics that make up the rest of the book.

I'm not an expert in this field, and I haven't read the whole book, but I sampled every chapter. The clarity of the text and the excellence of the dia-

grams impressed me. I also liked the editing and the formatting, with one exception. The publisher chose to indent every paragraph, except for the first paragraph in any first-level or second-level section. This makes passages that contain short paragraphs, centered formulas, and small diagrams difficult to scan.

I especially liked one feature of the book. Every few pages the authors have included boxes containing "points to remember." They have collected these into a nine-page checklist for system design at the end of the book. The checklist includes the chapter and section of each point, so it functions as a high-level outline of the main topics of the book.

If you aren't doing high-speed digital design now, chances are that advances in technology will soon move you into that bracket. Read this book now and avoid grief later.

Intel's SL Architecture—Designing for Portable Applications, Desmond Yuen (McGraw-Hill, N.Y., 1993, 345 pp. plus diskette; \$39.95)

In one of my early columns (April 1987) I looked at the then current state of manuals. We've come a long way since then. In those days a microprocessor supplier could only dream of putting a book of this quality into the hands of designers. Now they not only do so, but designers willingly pay \$40 for it. The effective use of outside publishers has been a large factor in making this happen. The book combines the inside knowledge of a senior Intel applications engineer with the publication expertise of a major publisher.

Yuen's book deals with how to use Intel's 386SL and 486SL CPUs and the companion 82360SL peripheral controller. Intel designed these chips to facilitate the design of portable computers. Their key feature, system management mode (SMM), addresses the central issue in this field, battery life.

Yuen takes you methodically through all aspects of designing with these chips.

It's like a complete collection of application notes with a coherent organization and a stronger than usual emphasis on principles. The associated diskette contains all of the sample programs. It even contains a debugging program and all of the register configuration files you need to use it.

If you are going to design with these chips, you need this book.

Guest book reviews

Our sister publication, *IEEE Software*, regularly solicits book reviews from a large pool of computing professionals. Recently, their backlog has grown. Many of these reviews are of books that *IEEE Micro* readers will find interesting. To help these reviews reach the public in a timely manner, we've included two of them here. If you have friends who read *Software* and don't usually receive *Micro*, I hope you'll show them a copy. Be sure to point out to them that subscriptions to *Micro* remain a great bargain.

Software Engineering: A Programming Approach, 2nd ed., Doug Bell, Ian Morrey, and John Pugh (Prentice Hall, Hertfordshire, England, 1992, 338 pp.)

Reviewer: J.E. Jordan, National Research Council, Ottawa, Canada

This easy-to-understand, nonmathematical overview of software engineering targets undergraduate students and software practitioners who wish to keep abreast of developments in the field. The book is a good example of the adage, "Small is beautiful." Concisely written in just over 300 pages, it is available in inexpensive paperback format but provides an extremely well-written, enjoyable, and readable treatment of the subject. Indeed, the authors succeed in conveying more usable information than many other more comprehensive, multivolume treatises.

The chapter on formal methods is a good example of a comprehensible treatment of a difficult subject. All of this should be welcome news to those

who want to learn important concepts in the field but who have limited time and money.

The material is aimed at satisfying the requirements of CS14: Software Development and Design in the ACM curriculum. It involves traditional techniques of software development with an emphasis on programming language and graphical methods.

This second edition includes material on new techniques such as object-oriented programming, parallel programming, formal verification, and issues of programming "in the large." The book's main strength is that it provides a meaningful look at a number of design methods and programming paradigms. The section on design is particularly worthwhile, illustrating functional decomposition, data decomposition, and object-oriented approaches. The comments on programming paradigms, also well-written, detail the different languages and philosophies available to developers.

Though quite interesting and relevant to software engineering, the final section on implementation is more of a grab bag, including software tools, validation and verification, fault tolerance, and programming teams.

While software development management is not covered extensively, the last chapter on programming teams does discuss one aspect of this topic. The book emphasizes technical issues such as programming languages and programming environments, which are probably appropriate for an introductory text for use in a computing science curriculum.

I recommend this book as an excellent introduction to software engineering and comparative programming language studies suitable for undergraduate students or practicing programmers who wish to keep current with recent developments.

Statistical Methods for Testing, Development, and Manufacturing, Forrest W. Breyfogle III (John Wiley

and Sons, N.Y., 1992, 516 pp.; \$64.95)

Reviewer: John W. Horch, The Horch Company, Madison, Ala.

This is the statistical reference book I've been waiting for. It is full of examples, contains simple text (to the extent possible for this topic), and includes a guide to use of the material. The statistics are mathematically correct, as I would expect. But the real value of the book is in the discussions of appropriate application of the statistics and the recognition of recent work in process improvement (Taguchi, Motorola, and others).

Before the actual text begins, Breyfogle guides readers toward their special interests. For example, the first table defines 30 or so areas of interest and the chapters dealing with them. As a software practitioner, I followed the directions and had great success when I was led to seven chapters that deal with the kinds of statistical methods applicable to software activities.

Breyfogle's second goal was "to make the topics practical to such an extent that this reference guide would become worn out." I believe this will be the case. Especially now as statistical methods are being bantered about in the software world, this book will be tremendously valuable to those who are charged with applying statistics to their work.

Achievement of the third goal, "to sell employees and all levels of management on the power of wisely applied statistical concepts," remains to be demonstrated. Readers may get a better understanding of their application with this book; however, wise application may be less achievable. My experience is that each "new" technique is adopted without much thought to the reason for the adoption. We need to be selective about the applications rather than blindly going forth, having limited success, and losing management's commitment because we lack useful results.

After this obligatory discussion, Breyfogle gets right to the point: "de-

fine the problem or question you want to answer." For a requirements bigot like myself, this is good news.

The heart of the book is the second section. The author reiterates the need to define the problem and then describes the analysis being applied. A later "do it smarter" section offers shortcuts that reduce time and effort, and help direct the analysis to specific customer needs.

A final section on real-world situations helps the reader begin to see how the thoughtful and appropriate application of statistics can add significantly to the quality of the testing, development, or manufacturing process.

While this may not be the book that provides "everything you ever wanted to know" about statistical methods, it is a most useful guide, the best I've seen. I expect to wear my copy out.

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 183 Medium 184 High 185

Search for Authors

IEEE Computer Society Press offers

- ★ International promotion and distribution
- ★ Prestige
- ★ Competitive Royalties
- ★ A supportive development environment
- ★ High-quality production

We publish monographs, collections of readings, and tutorial texts for professional and educational markets as well as for our 100,000+ membership.

Talk to us.

CONTACT: Henry Ayling, Editorial Director
h.ayling@compmail.com



**IEEE COMPUTER SOCIETY
PRESS**

10662 LOS VAQUEROS CIRCLE
LOS ALAMITOS, CA 90720

(714) 821-8380

Fax: (714) 821-4010



THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.

Software Report



David K. Kahaner

US Office of Naval

Research, Far East

kahaner@cs.titech.ac.jp

Completely automated assembly

Now that automating production lines has become commonplace, the next logical step is to completely automate assembly lines. Japan's semiconductor industry is aggressively pursuing this trend, as are segments of its household electrical goods industry. Small- and medium-size assembly lines and those where many models are assembled in small lots have yet to implement full automation. But even these are feeling the pressure to climb on the bandwagon.

Assembly lines that have already begun to implement automation, moreover, find themselves in various stages according to the degree of automation they employ, their flexibility, and the feedback they give to product design. Lines where the assembly of existing products has been converted from hand assembly to robot assembly we call "first-generation" automated lines. Those that are automated to the extent that some product design changes can be made we refer to as "second-generation" lines. Lines having automation features that impact broadly on product design then are "third-generation."

All assembly lines, no matter what their generation, are subject to a great many limitations and conditions when they are built. Here we examine the progress made by a range of leading Japanese industries—makers of everything from caterpillars to computers—to gauge current progress toward fully automated assembly. Included in our survey are automated assembly lines at Daikin Industries, Gunma Nippon Electric, and Seiko Epson. Of these lines, only Epson's printer assembly line had been previously automated; until very recently the other lines all employed manual assembly.

To get a better feel for these trends, we will look at the developments at one of these com-

panies, Gunma Nippon Electric, to see what a fully automated assembly line entails. The basis for this analysis comes from an article in *Nikkei Mechanical* (June 29, 1992).

Automation-oriented design techniques

Much of the credit for the traditional cost competitiveness and high quality of Japanese manufactured goods belongs to the skillful and pervasive automation of its production lines. More recently, automation there has become a critical component in dealing with labor shortages, demands for improved work environments, and the need to implement companywide computer-integrated manufacturing (CIM).

Discussion currently focuses on automating the assembly line. This emphasis is quite distinct from automating fabrication lines—a simpler case where systems need only be built to automate the conveyance of work to numerically controlled machine tools and the like. In automating assembly lines, robot hands and jigs must be adjusted to handle each part, greatly complicating parts feeding and line control.

Now that industry has made so many strides with factory automation, equipment, production control, and computers, building automated assembly lines no longer seems unusual. Until now, however, either high-volume production operations, such as for electrical goods, or products having relatively simple structures, have accounted for most of this assembly automation. These applications have not, however, spread through the entire manufacturing industry, as has been the case with fabrication line automation.

Three technological advances. A broad range of applications has advanced dramatically by automating their assembly lines. Included are

everything from commercial outdoor air conditioners (Daikin) and hydraulic shovels (New Caterpillar Mitsubishi) to personal computers (Gunma Nippon Electric), and computer printers (Seiko Epson).

In these lines, products that come in many different models must be assembled in small- and medium-size lots and require high-precision assembly, making automation difficult. Labor shortages and demands to implement computer-integrated manufacturing have increased the pressure to automate these lines.

Assembly automation applications have broadened, however, with the recent advances in production technology, including robots and sensors, component feeding, line control, and production management. The new printer assembly line that Seiko Epson began operating in April 1992 provides one good example of these advances.

Seiko Epson had successfully automated its assembly operations in 1984. To take advantage of subsequent developments in automated assembly technology, the company needed to refurbish its line, a decision that also greatly influenced the design of the company's new printer models.

What, specifically, are these advances in automated assembly technology? A survey of the latest automated lines in Japan noted three major areas of development:

- Positioning technology, which is fundamental to robotics,
- High-flexibility line construction technology that can cope with the mixed-flow production of multiple product models, and
- Product design technology aimed at assembly automation.

Multipurpose use of visual sensors. Common to all assembly automation is the problem of positioning. Automation would be a relatively simple procedure if parts could be neatly lined up and carried along, but

the size and shape of parts often makes this impossible. Related problems include the need to reposition parts such as the large metal plates used in hydraulic shovels from which arise plate fabrication irregularities. Deformations in parts as with personal computer printed circuit boards can also create difficulties. Frequently, an assembly point will be out of position even though the end or edge of a part is properly placed.

***Automation
would be
relatively simple
if parts could be
neatly lined up
and carried along.***

The first step in positioning is to develop jigs that match the parts. After taking up a part, a robot first must place it on or in a jig before assembly can commence. Then the robot must grip the part again. The jig determines the position and attitude of the part. With large metal plates, positioning can only take place after special jigs have been developed to hold the plates in the proper position for assembly.

The shape of a part can make positioning with a jig difficult. Sensors sometimes must be used to correct the movement of the robot, but accurate sensor detection is often very difficult. Printboard warping provides a typical example. After running into problems with printboard warping, Gunma Nippon developed a triangular measuring optical sensor for use in detecting the height of three points on a printboard. A two-dimensionally curving warp can be represented by three

points, so the company had to come up with innovations in measurement point selection and interpolation techniques.

Visual sensor applications have now become somewhat standard. Visual sensors can detect the positions of a great many parts and are effective in implementing mixed-flow production operations. In addition to detecting positions, these sensors can also detect the shapes of parts, making them most useful in product quality control.

To be sure, the positional correction precision of a visual sensor is not terrific. When we factor the mechanical error of the robot together with the limitations of image processing resolution, errors of several hundred microns can and do develop. More innovation for high-precision assembly soon becomes necessary.

In Daikin's outdoor air conditioner unit assembly line, visual sensors combine with remote center compliance (RCC) mechanisms to perform high-precision nesting operations with compressor parts. The compressors used by Daikin are scroll-type compressors that require approximately 10- μ m precision in operations such as nesting crankshafts into bearings. Visual sensors alone do not afford this degree of precision, so RCC mechanisms attached to the robot hands must absorb the remaining error. The RCC mechanism contains an internal spring mechanism, enabling it to search for accurate nesting positions by the deflection of the spring from work forces.

Movable jigs and sensors simplify production control. The assembly lines at New Caterpillar Mitsubishi and Daikin are designed to cope with the demands of multiple-model, mixed-flow production. To increase flexibility, the assembly operations there use movable jigs to implement controls that accord with part shape. Visual sensors also play an important role.

In a mixed-flow assembly line, information on product models must be controlled so that the line operation

can be adapted to match the model moving on the line. Basically a computer working over a LAN handles this information control, but simpler and more reliable methods are needed.

Recognizing this need, Daikin has equipped its pallets with memory cards to achieve "object-information integration." The memory cards contain model information such as parts dimensions, detection standards, and other vital data that can be read at each assembly stage. This way, the operation can be changed easily to accord with the model.

Daikin employs its own methods for line control and parts supply. It was a sequence-synchronized production method for ordering the supply of parts assembled in the main work flow, and attach numbers to the work to indicate production order. Parts then can be ordered from the automated warehouse based on these numbers.

Focus on modular design. If we approach assembly automation merely from the perspective of production technology, however, problems involving equipment costs and reliability soon arise. Consequently, it has become standard practice to reevaluate product designs and to implement design features compatible with automated assembly operations. We can then simplify assembly and enhance operational reliability by, for example, orienting all the assembly steps in one direction, or employing connection techniques amenable to automation.

To maximize the effectiveness of these measures, we would prefer to have "new lines for new models." That way we could develop parts concurrently with line construction. Investment efficiency and problems arising from the product model change cycle often force us, however, to automate a line for an existing model.

Looking at recent design trends, the modular approach adopted by Daikin on its outdoor air conditioner units is noteworthy. For design purposes, product structures are divided into a number of modules. Each module is

assembled on a subline, and the assembly operations that are not amenable to automation are concentrated in the final assembly line. Automation rates in the total assembly process rise easily because each module is designed for compatibility with automated assembly. Even automobile manufacturers are attempting to modularize their products so that they can automate the final assembly process.

These product design techniques

***Investment
efficiency and
model change
cycles often force
us to automate
lines for existing
models.***

should change as assembly automation technology continues to progress. The printer line at Seiko Epson provides an extreme example. For some time, Seiko Epson has tried to implement unidirectional parts assembly to facilitate assembly automation. Unidirectional assembly, however, translates into complex parts shapes and higher fabrication costs—the total manufacturing cost escalates even though assembly costs are low.

With robots becoming so highly functional, inserting a part diagonally no longer presents the challenge it once did. The best policy then is to calculate parts fabrication costs and assembly costs to arrive at the most favorable method of assembly. Seiko Epson designed its new printers in the context of total manufacturing cost and constructed its assembly line accordingly.

Improving the automation rate while developing ways to handle multiple-module, mixed-flow production becomes the next task. The mixed-flow production approach is effective in holding down equipment costs and coping with production fluctuations.

The key to implementing assembly automation is the development of automation-compatible designs. This design-oriented approach is a new concept, but one with great potential. Developing programs to coordinate design and production now takes on increasing urgency.

To get a clearer picture of all that fully automating the assembly line involves, let's take a nuts-and-bolts look at one of these companies.

Automated computer assembly

Gunma Nippon Electric is the main development and production center for personal computers carrying the NEC name. Gunma NEC built an automated assembly and inspection line for desktop PCs in November 1991. These computers represent more than 10 million users. Some 1.17 million units were produced in 1991. But even at NEC, which has more than half the domestic share in this market in Japan, computers were assembled manually until 1991.

"We began planning the automated assembly line in 1987," says Masaki Takahashi, manager of the Systems Division of the CIM Systems Department. "But the plans were delayed when notebook PCs hit the market and raised the specter of declining desktop demand." Apparently, the biggest obstacle then was return on investment. But what about problems with production technology? Looking at the line, we see that a number of innovations have been implemented.

Moving ahead with production automation. Gunma NEC is working with other parts of NEC to implement computer-integrated manufacturing. They aim to tie production and supply

to market fluctuations. When a product sells well or poorly, the number of PCs produced should rise or fall accordingly.

Toward this end, Gunma NEC wants to shorten what it calls "production multiplier lead time." This refers to the time required to double the number of PCs that the company plans to produce. The company wants to take the time to fix the production plan and, as nearly as possible, set it to the actual number of product days. All the factors involved, from parts procurement to production systems, must be reevaluated to achieve this goal.

Maintaining some degree of overstocking will handle the parts problem. For production systems, excess production capacity should be maintained representing 120 percent of the average number of products shipped. Even these steps, however, cannot fully absorb fluctuations in PC demand. Thus they modified the single-shift production operation, with normal 8-hour shifts, to handle two or three shifts.

However, a problem arises—retaining skilled personnel to work the shifts. Some 15 skilled workers are required to cover a manual assembly line, so 30 must be retained to handle a double shift. But how can this be done when the extra workers are only needed during periods of increased production? The problem will be solved if production can be automated and unmanned production implemented. Gunma NEC is moving ahead with production automation. In 1988, the company automated packaging, and in 1989, they automated the printboard assembly and inspection operations. In late 1991, Gunma NEC built a robot line that automated the final assembly and product inspection stages.

Reducing skilled workers by one third. The robot line has nine assembly operations covering a length of 50 meters, and 12 inspection operations that cover 30 meters, excluding the running test room. Four of the robots employed are six-axis vertical articu-

lated models, eight are three-axis transverse models, one is a two-axis transverse type, and two are horizontal articulated types, for a total of 15 robots. The total investment was approximately 500 million yen.

Of all the operations done on this robot line, only two assembly operations and three inspection operations are performed manually. The two manual assembly operations both involve hooking up cables. The cable assembly operations involve two as-

***Stringing a
pliable object like
a cable is one of
the most difficult
jobs for a robot.***

pects, namely plugging in the connectors and stringing the cables. Stringing a pliable object like a cable is one of the most difficult jobs for a robot to attempt.

Though this robot line has not completely eliminated manual labor, it has reduced the number of skilled on-site workers required to just five—a 300 percent reduction over the manual line. At the risk of oversimplifying the situation, this automation has made it possible to take the same number of skilled workers from the manual line and spread them over three shifts on the robot line. The start-to-finish tact time has been reduced by 20 seconds, from 77 seconds on the manual line to 57 seconds on the robot line, making it possible for the robot line to turn out 450 PCs in an 8-hour shift.

Components positioned with special jigs. The robot line employs almost the same assembly sequence as does the manual line. The components comprising the PC are very few, including only the baseplate, U-shaped cover,

front mask, rear cover, motherboard (on which the processor is mounted), floppy-disk drives, and a few cables. And since the components making up the PC are so few, the sequence in which they are assembled does not need to be changed.

In the assembly operation, the base is first attached to a jig pallet, and the main printed circuit board (motherboard) is installed. The motherboard is screwed into place in the next operation. Next, a chassis for mounting the floppy-disk drive is installed and the drive is mounted in it. A cable between the drive and motherboard is connected by hand. The expansion cage is then built in, and screwed to the floppy-disk drive chassis and expansion cage. A cable is then manually connected to the expansion cage, and the front mask is attached at the same time. The rear and top covers are then attached and screwed into place to complete the assembly. Note that all component positioning occurs simultaneously before any are mounted. The components are not small enough to be supplied by a vibrating parts feeder, so they are placed behind the robots on a pallet.

The robots take components from the pallet and place them in a special positioning jig. The jig positions the components with air-pressure cylinders. From the standpoint of tact time, the reasons for adopting a disadvantageous positioning method are as follows. Using a pallet that can carry the components and perfectly position them involves prohibitive pallet fabrication costs. Placing the components on the pallet then would also be problematic, adding to component costs. Also, most of the components used in a PC are fabricated from sheet metal. Compared to machined parts, the precision of these fabricated units is poor, so automated assembly cannot be done if the positioning is sloppy.

Handling printboard warp. Positioning components with jigs does not solve all the problems. Screw-hole

positioning precision and component warping difficulties remain, for instance. To deal with such problems, one can either measure the screw-hole position or warp and adjust the movement of the robot accordingly, or one can implement more exacting component precision.

The measurement approach requires longer tact times. Using more exacting precision requires higher component costs. These problems led Gunma NEC to reexamine the criteria for all of its components. For a good example, let's look at what they did with the motherboard, a big square printboard measuring 305 mm². Conventionally, hole positions had been marked sequentially from one edge of the board, which produces decreasing precision as the distance from the edge to the hole increases.

In securing a printboard, the important consideration is the relative position of one hole to another, much more so than the relative position between a hole and the edge. This being so, beginning with the PC9801 FA series that went on sale January 1992, Gunma NEC changed the design of the printboards to indicate the positions of holes in terms of one standard hole located near the centerline, eliminating the need to measure hole positions.

Tinkering with the standard positions, however, would not resolve the problem of motherboard warping. Almost all the mounting and soldering of electronic components to the printboards has now been automated. The soldering involves wetting the printboards with molten solder; the heat from this process unavoidably produces printboard warping. Since the motherboard is so large, a small angle of warp can produce displacements measured in millimeters.

For this reason, when printboards are positioned at Gunma NEC, the printboard warp is measured with optical sensors using triangulation. Since there is no time to take measurements over the entire surface of the board,

three sensors measure the warp at three points simultaneously. For this reason, mistakes are sometimes made in detecting warp. After the motherboard has been installed, it is inspected to see whether or not it is in the right position.

Manual assembly also possible. Unfortunately, a robot line is longer than a manual assembly line. At Gunma NEC, the conventional manual assembly line is 40 meters long, divided roughly equally between assembly and inspection stages. The robot line, how-

**Research has
already begun on
ways to
automate the
assembly of
notebook
computers.**

ever, is 80 meters long, with 50 meters for the assembly stages and 30 meters for the inspection stages. In general, when production is roboticized, the line becomes longer largely because robots have a hard time doing more than one thing. At Gunma NEC, each robot is set up to install two components. Even so, the robot line is twice the length of the manual line in the interest of maintenance. "We allowed plenty of extra space so that it would be easy to perform maintenance on it and revert to manual assembly in the event of a breakdown," says T. Ono, manager of the Production Technical Division.

Easier to add components. Their limited adaptability when models change also creates problems for automated robot lines. A robot can assemble all kinds of components if the

program that controls the robot's actions is modified and the robot hand is adjusted properly. If the number of components grows, however, a robot line is not very adaptable at all. To keep the tact time short, more robots must be installed, an approach that is both expensive and time consuming. If the tact time can be lengthened, the problem remains of how to supply the components when the number of components assembled by each robot increases. The robot line at Gunma NEC is built so that the number of components assembled by any one robot can easily increase, making the line highly adaptable to production model changes.

Studies are underway to find ways to eliminate the cables that must be manually hooked up. If successful, this research should make it possible to achieve completely unmanned automation of assembly lines. Research has already begun on ways to automate the assembly of notebook computers, which are much smaller than the desktop PCs and hence much harder to assemble automatically. Gunma NEC intends to employ robots in ways that will make it possible to adjust its production volume to demand vicissitudes.

[David Kabaner is on assignment with the US Office of Naval Research, Far East. His comments are his own; they do not express any official policy.]

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 195 Medium 196 High 197



New Products

Send announcements of new microcomputer and microprocessor products to
Managing Editor, IEEE Micro, PO Box 3014, Los Alamitos, CA 90720-1264.

Joe Hootman

University of
North Dakota

CAD tools

VHDL package speeds FPGA design

Release 1.21 of the Complete Optimization/Retargeting Environment (CORE) offers FPGA designers a top-down, three-step design approach based on the VHSIC Hardware Description Language (VHDL). Users enter designs in VHDL, then CORE translates and optimizes them for the target FPGA device. CORE can be seamlessly integrated with IEEE 1076-compliant simulators, along with VHDL-producing systems such as i-Logix, Vista, and Ascent.

This release includes specialized optimization techniques to support Actel Act 3 and Altera Max 5000/7000 series devices. CORE has also been ported to the Hewlett-Packard 700 series hardware platform and the Motif Windowing System. *Exemplar Logic; from \$8,000 (CORE pre-seat cost); upgrades free with maintenance contracts.*

Reader Service No. 10

Module supports XC4000s

An FPGA Foundry addition supporting Xilinx XC4000 devices lets designers use a single tool set and take advantage of the Timing Wizard module to set timing parameters and operating frequencies at the beginning of design. FPGA Foundry also works with other FPGA vendors and architectures and integrates into existing CAE environments. The XC4000 release includes fast carry logic, wide edge decoders, RAM, partially or fully placed and routed hard macros, guide files, and a graphical logic block editor. *NeoCAD; from \$4,995; \$4,500 (PC upgrades).*

Reader Service No. 11

FPGA system targets PALs

PAL users can convert to FPGA design methodologies with Designer, a "purchase-once" FPGA design system that supports the company's

devices under 2,500 gates. Designer includes the Action Logic System and choice of design kits for PC 386 and 486 hardware platforms. The three design kit configurations support OrCAD, Viewlogic, and EDIF netlist interfaces and include device macro lines and simulation models. For designs needing higher device densities, Designer Advantage accommodates Act 1/2/3 devices up to 10,000 gates and Sun-4, Sparc, Sparc 2, 386/486, and HP series 700 platforms. *Actel; \$995 (Designer), \$495 (Designer Advantage, current users).*

Reader Service No. 12

Design Center with PSpice

Available for the Quadra, Powerbook, Mac IIvx, Performa, and Macintosh workstations with math coprocessors and 2-Mbyte RAMs is a CAD/CAE system called Design Center. With this design environment users can simulate analog, digital, mixed analog/digital circuits with PSpice at all levels of the design process and analyze graphical waveforms.

Design Center also supports the HP Apollo 9000 Series 700 workstation, which, according to the company, permits circuit simulations to finish 12 times faster than on a 386/33-MHz PC and twice as fast as on a Sun Sparc 2. *MicroSim; \$4,950 (Macintosh version), \$17,900 (Series 700 version).*

Reader Service No. 13

FPGA, PLD software combined

The XACT Base Development System combines FPGA and existing programmable logic development software in support of 3,000-gate devices. For FPGAs, the software includes an interface to compile OrCAD or Viewlogic design and offers incremental FPGA design so engineers can quickly make changes. An XDelay static timing calculator, download software, and

parallel download cable help verify designs. For ELPDs, the system adds Palasm-compatible equation files for design use. *Xilinx*.

Reader Service No. 14

Embedded RISCs to access VxWorks

Designers of high-performance embedded applications based on the AMD 29000 processor will soon be able to access the VxWorks operating system and development tool suite for embedded control. VxWorks lets users develop and execute complex real-time and embedded applications with a Unix cross-development package that networks designs, testing, and debugging tools with target hardware. *Wind River Systems*; 4Q93 availability.

Reader Service No. 15

Tool aids three-layer routing

Microroute, an ASCII-format placement-and-routing solution for three-layer metal, mixed block and cell designs lets users choose either automated or manual approaches. A global router handles corners and intersections across the entire design, while an N-layer detailed maze router produces dense results in specific areas. Designers can create files by writing directly from their system or through EDIF and GDSII Stream interfaces. *Mentor Graphics*.

Reader Service No. 16

IEEE P1284 ECP design kit

System designers wishing to design motherboards and add-in cards to support the IEEE P1284 Extended Capabilities Port protocol can use the ECP/EPP Super I/O Design Kit. ECP provides a high-speed bidirectional port that is backward-compatible with existing cables and connectors. The port should improve the performance and ease of use of parallel peripherals. The kit provides documentation, software, schematics, and a demonstration board. *Standard Microsystems*.

Reader Service No. 17

MCM testing, diagnosis

Multichip module and PCB manufacturers can test and diagnose products with the MCM Probing System, which merges advanced station control and CAD navigation software with the Micromanipulator precision probe placement platform. The IDE-compatible, touch-sensitive probing system allows several probes to be activated simultaneously. It displays or stores real-time waveforms from sampling oscilloscopes and logic analyzers within the DSO and XLA Tool windows using IEEE 488. Users can automatically track full logging of the diagnostic session while accessing interactive control of features via mouse-driven, pop-up menus. The modular system provides automated operation drawing from either IC- or PCB-based design and layout data. *Schlumberger Technologies, ATE Division*; from \$95,000.

Reader Service No. 18

Desktops can access CADAM AEC

Engineers with PS/2s can access CADAM AEC tools to design facilities from preliminary layout through the construction phase by adding the Personal/370 Coprocessor. The turnkey system uses a mainframe VM system card that can be inserted into a PS/2 running OS/2. Since it runs host CADAM V3R2MO, this platform has full functional and data compatibility with a mainframe seat; no retraining is required. *Integrated Systems Technologies*.

Reader Service No. 19

DSP systems

Generate C code with Windows

Version 1.0 of the Hypersignal-Windows Block Diagram object-oriented simulation program for signal processing lets users generate C code to run algorithms outside Block Diagram. Block Diagram implements a DSP design by proving the algorithm, debugging it, and setting up what-if situations in a visual environment. The code gen-

erator then produces C code for the algorithm, which may be compiled with the DSP chip manufacturer's C compiler. This lets the program run in real time on a DSP. *Hyperception*; \$2,995 (code generator), \$1,995 (Block Diagram).

Reader Service No. 20

Integrate DSPs/gate arrays

Beginning a new line of standard DSPs embedded in gate arrays is the 100-/144-pin thin SQFP TEC320C25A. System designers can use the chip to develop customized DSP solutions that get to market quickly. The 15K-gate, 60-MHz, 15-MIPS TEC320C25A integrates two standard, high-volume devices onto one chip as an enhanced version of the 16-bit, fixed-point TMS320C25 and an array from the 0.8-micron TGC1000s. Designers can use the TGC1000 library of gate arrays to integrate various logic functions with the DSP core. *Texas Instruments*.

Reader Service No. 21

PC voice recognition

Designed to increase personal productivity by adding a voice command interface tied to keyboard and mouse macros, Voice Blaster system runs on Intel-based personal computers using DOS or Windows 3.1. The voice recognition system includes a toolbox of revised programs and utilities for recording, editing, and playback, as well as voice annotation software that adds a user's own recorded messages to documents. A high-fidelity headset with microphone and speaker connects to a user's computer via the parallel port. Though fully functional on a 286 with 640-Kbyte RAM, in the presence of EMS memory, Voice Blaster software will automatically load as much of itself as possible in high memory. *Covox*; \$119.95.

Reader Service No. 22

Multimedia PC audio decoding

Two single-chip audio coder/decoders address the needs of multimedia

personal computers, providing stereo, 16-bit audio. The 68-pin PLCC-packaged CS4248 and CS4231 codecs are pin-compatible with the Analog Devices AD1848 and come with Windows-compatible drivers.

The CS4248 ADC/DAC uses proprietary delta-sigma conversion techniques to code and decode audio signals, making CD-quality sound available. It includes an 8-bit parallel ISA/EISA interface, analog mixers, antialiasing and reconstruction filters, and simultaneous capture and playback capabilities. The enhanced CS4231 version offers 4-to-1 adaptive differential pulse code modulation compression/decompression and supports different data formats. *Crystal Semiconductor; from \$35 each (1,000s).*

Reader Service No. 23

ADCs speed at 1-Msamples/s

LTC1273, LTC1275, and LTC1276 300-Ksample/s analog-to-digital converters contain a precision reference, a high-speed sample and hold, and an internal clock. Typical signal-to-noise distortion on the 24-pin narrow DIP or 24-lead SOIC chips is 72 dB for 10-kHz inputs and 70 dB for 100-kHz. The LTC 1273 runs on one 5V supply and converts 0V to 5V inputs. The LTC1275 and LTC1276 run on $\pm 5V$ and convert $\pm 2.5V$ and $\pm 5V$ inputs.

The LTC1196 and LTC1198 1-Msample/s, 8-bit ADCs come in S0-8 surface-mount chips and offer 600-ns conversions. The switched-capacitor, successive-approximation chips include 100-ns sample and hold on chip; both operate from 2.7V to 6V power supplies. *Linear Technology; from \$14.09 (300-Ksample versions, 100s), from \$2.37 (1-Msample versions, 1,000s).*

Reader Service No. 24

Mac-based data acquisition

The 50-Ksample/s MacScope system provides hardware and software for educational, research, and industrial applications where multichannel data acquisition is required. The System 7-

compatible product running on a Macintosh Plus, SE, or II also provides data analysis such as Fourier transforms. Features include a mouse, pull-down menus, scroll bars, and data file storage with screen-printing options. *World Precision Instruments.*

Reader Service No. 25

Acquire data with Windows DDE system

A recent version of Snap-Master for Windows 3.1 lets engineers and scientists acquire, display, analyze, and output data with Dynamic Data Exchange support. Snap-Master version 2.0 integrates sensors, transducers, and signal conditioning. Features include context-sensitive on-line help, zooming and panning for large files, multiple cursors, and event markers.

Users can transfer data to a spreadsheet for real-time trend analysis and report generation while Snap-Master acquires information in the background. Version 2.0 requires a PC or PS/2 and a 4-Mbyte memory and comes in three stand-alone modules that also work as an integrated package. *HEM Data; from \$495 (modules), \$1,985 (package).*

Reader Service No. 26

VME boards provide DSP subsystems

According to the company, its 200-MFlops/1-Gops 6U VME boards based on the TMS320C40 DSP let embedded systems designers reduce development time by 50 percent. Both CV2 and CV4 boards combine DSP, array processing, and parallel processing capabilities of the C40 with I/O and standard interfaces. Debuggers, libraries, compilers, assemblers/linkers, and other software tools complete the systems.

Each board can be used with TIM-40 modules to optimize designs and can be configured with eight processors, large DRAM arrays or fast SRAM arrays, and special-purpose modules such as video capture and SCSI. *Spectrum Signal Processing; from \$9,100.*

Reader Service No. 27

Voice processor boasts open architecture

The BT-IV digital voice processor delivers high-fidelity speech, built-in redundancy, and digital-controlled components for application program control provided by micro, mini, or mainframe computers. The tower or rack-mount BT-IV connects to value-added services such as ISDN and world standards. A switch-host feature enhances fail-safe applications. *Perception Technology; \$1,500 per channel.*

Reader Service No. 28

Communications/displays

Ethernet link added to multiscreen display

The Media Wall multiscreen display system for control room and simulator applications features an Ethernet link to Sun, SGI, HP, DEC, IBM, and other workstations. In normal room lighting, Media Wall displays graphical or photographic information in an array of 144 monitors or projectors controlled by one computer. The displays can be placed in one line or circle, a rectangular matrix, or other shapes for specific requirements. *RGB Spectrum.*

Reader Service No. 29

VGA monitor survives in harsh areas

Designed for extreme conditions of temperature, humidity, dirt, shock, and vibration, a slim flat-panel line of VGA monitors displays graphics in active matrix color LCD or monochrome EL screens. Available with infrared touch systems and touch mouse features, the Seal Touch monitors weigh about 25 pounds and come in stand-alone or OEM module versions. *Lucas Deeco; from \$3,650; delivery 60 days ARO.*

Reader Service No. 30

X terminal promises 1,200×1,024 resolution

A RISC-based, color flat-panel X Terminal, the XfaceC, features 70,000-

Xstone performance and 1,280x1,024 resolution. The 25-MHz system designed to work with most platforms and in multiple-host environments offers 256 simultaneous colors on a 13-in. LCD screen and an X server accelerator to deliver fast screen updates and offload bitblt, fill, and arc primitives to hardware. *Japan Computer Corporation*; \$10,000.

Reader Service No. 31

Communications programs added

Three communications packages support DOS, Microsoft Windows, and Windows NT.

Version 3.4 of DynaComm/Elite for Windows-based PC-to-mainframe connectivity offers concurrent 3270 and LU6.2/APPC communications. Version 2.2 of 3270/Elite Plus is a multisession, DOS-based, PC-to-mainframe connector for SNA LU2 environments.

The third package, the 3270 Emulator, connects over IEEE 802.2 networks and supports one host display session plus copy and paste functions. The simplified 3270 PC-to-mainframe communications program ships with each copy of the 32-bit Windows NT operating system for PCs. *Network Software Associates*; \$395 (Elite packages); corporate and network licenses available.

Reader Service No. 32

Displays use PCMCIA memory

Max/it displays come with Options/Open to use PCMCIA flash memory technology in general-purpose alphanumeric terminals. Two protocol-specific/ANSI emulation terminals, the Max28 and the Max1120, are designed for the Uniscope market and require one PCMCIA card per site. The Max10BT includes a 10BaseT and an AUI Ethernet port for plug-in LAN connection and works with TCP/IP or Novell Netware protocols. The Max6 multisession ASCII/ANSI/graphics terminal completes the family. *Link Technologies*; from \$649.

Reader Service No. 33

ATM hub delivers LANs/WANs

An Apex Asynchronous Transfer-Mode switching hub for corporate networks supplies both pure cell switching and adaptation switching interfaces for non-ATM traffic in one platform. ATM supports very high transmission speeds for integrated data, voice, and video. Apex interconnects LAN hubs with ATM or Ethernet interfaces; transports circuit-switched data, voice, and video; switches frame relay and X.25 traffic; and transmits HDLC and SNA/SDLC-framed information within one backbone network. *General DataComm*; from \$50,000 to \$125,000 (typical system configurations).

Reader Service No. 34

IRMAs support Windows NT, OS/2

IRMA Workstations now support Microsoft Windows NT and OS/2 2.0 Presentation Manager with a 32-bit host access communications package that integrates the IBM SNA network. Both versions can access DFT, SDLC, X.25, and IEEE 802.2 environments over token ring or Ethernet. They emulate the IBM 3270, Logical Unit 6.2, LU0, and LUA Advanced Program-to-Program Communications. Features include a keyboard editor and QuickPad for frequently used keys. The NT package can process 10 concurrent 3270 sessions. *Digital Communications Associates*; \$495 each version.

Reader Service No. 35

Multiplatform X.5 servers

A line of X server software solutions based on release 5 of the X Window System from MIT offers Unix connection to Apple Macintosh, NextStep, and MS Windows computers. Known as both X11R5 and eXodus 5.0, the multiplatform products support X font formats, networked X font servers, and the enhanced DECwindows, Sun Open Windows, and Motif. *White Pine*; \$295 each (Macintosh version), \$349 (NextStep); \$449 (MS Windows).

Reader Service No. 36

Simplify ASIC design

Eight FSB cells for T1/E1 applications in PCM multiplexing, switching, and transmission systems are blocks of application-specific logic that may be embedded into an ASIC chip. After being surrounded with user-specific logic, the cells provide a customized solution for a system design problem. By combining several functions and up to several thousand gates, system designers can quickly implement specific communication functions. *VLSI Technology*.

Reader Service No. 37

Wireless LAN connects PC shoppers

Small-business customers requiring simple, powerful computing solutions can purchase a wireless peer-to-peer LAN system called Advantage! Net. With built-in wireless communications and preinstalled application software, the hard-wired network alternative uses Windows for Workgroups linked to a wireless RangeLAN adapter from Proxim. A 25-MHz, 486SX-based, 4-Mbyte desktop and an 8-Mbyte, 66-MHz, 486DX2 minitower with 120-Mbyte tape backup make up Advantage! Net. Sold nationwide at 900 retail locations. *AST Research*; under \$2,000 (486SX/25), \$3,500 (486DX2/66).

Reader Service No. 38

Handheld protocol analyzer for notebooks

A PC-based protocol analyzer line communicates with a PC via a standard 4-bit or 8-bit parallel port and does not require a card slot. Designed to work with the low to medium WAN segment of data communication testers, the handheld Feline PS2002 ParaScope is housed in a 6.22x3.74x2.17-inch molded plastic case. It offers 19.2-Kbps RS-232 data monitoring and analysis. A PS6002 64 works with applications requiring 64K performance, while the PS6145 64M offers 64K performance with integral interfaces for RS-232, X.21,



June 1993 issue (card void after December 1993)

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone(_____) _____

Reader Interest

(Add comments on the back)

Readers,
Indicate your interest in
articles and departments
by circling the appropri-
ate number (shown on
the last page of articles
and departments):

150 151 152 180 181 182
153 154 155 183 184 185
156 157 158 186 187 188

159 160 161 189 190 191
162 163 164 192 193 194
165 166 167 195 196 197

168 169 170 198 199 200
171 172 173 201 202 203
174 175 176 204 205 206

177 178 179 207 208 209

Product Information

(Circle the numbers to receive
product information)

1	21	41	61	81	101	121	141
2	22	42	62	82	102	122	142
3	23	43	63	83	103	123	143
4	24	44	64	84	104	124	144
5	25	45	65	85	105	125	145
6	26	46	66	86	106	126	146
7	27	47	67	87	107	127	147
8	28	48	68	88	108	128	148
9	29	49	69	89	109	129	149
10	30	50	70	90	110	130	
11	31	51	71	91	111	131	
12	32	52	72	92	112	132	
13	33	53	73	93	113	133	
14	34	54	74	94	114	134	
15	35	55	75	95	115	135	
16	36	56	76	96	116	136	
17	37	57	77	97	117	137	
18	38	58	78	98	118	138	
19	39	59	79	99	119	139	
20	40	60	80	100	120	140	



June 1993 issue (card void after December 1993)

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone(_____) _____

Reader Interest

(Add comments on the back)

Readers,
Indicate your interest in
articles and departments
by circling the appropri-
ate number (shown on
the last page of articles
and departments):

150 151 152 180 181 182
153 154 155 183 184 185
156 157 158 186 187 188

159 160 161 189 190 191
162 163 164 192 193 194
165 166 167 195 196 197

168 169 170 198 199 200
171 172 173 201 202 203
174 175 176 204 205 206

177 178 179 207 208 209

Product Information

(Circle the numbers to receive
product information)

1	21	41	61	81	101	121	141
2	22	42	62	82	102	122	142
3	23	43	63	83	103	123	143
4	24	44	64	84	104	124	144
5	25	45	65	85	105	125	145
6	26	46	66	86	106	126	146
7	27	47	67	87	107	127	147
8	28	48	68	88	108	128	148
9	29	49	69	89	109	129	149
10	30	50	70	90	110	130	
11	31	51	71	91	111	131	
12	32	52	72	92	112	132	
13	33	53	73	93	113	133	
14	34	54	74	94	114	134	
15	35	55	75	95	115	135	
16	36	56	76	96	116	136	
17	37	57	77	97	117	137	
18	38	58	78	98	118	138	
19	39	59	79	99	119	139	
20	40	60	80	100	120	140	

SUBSCRIBE TO IEEE MICRO

All the facts about today's chips and systems

☐ **YES**, sign me up!

If you are a member of the Computer Society or any other IEEE society,
pay the member rate of only \$23 for a year's subscription (six issues).

Society: _____

IEEE membership no: _____

Society members: Subscriptions are annualized. For orders submitted March through
August, pay half the full-year rate (\$11.50) for three bimonthly issues.

Full Signature _____ Date _____

Name _____

Street _____

City _____

State/Country _____ ZIP/Postal Code _____

☐ **YES**, sign me up!

If you are a member of ACM, ACS, BCS, IEE (UK), IEEE but not a
member of an IEEE society), IECEJ, IPSJ, NSPE, SCS, or other
professional society, pay the sister-society rate of only \$42 for a year's
subscription (six issues).

Organization: _____ Membership no: _____

For information on airmail option, send fax request to (714) 821-4010.

☐ Payment enclosed *Residents of CA, DC, Canada, and Belgium add applicable sales tax.*

☐ Charge to ☐ Visa ☐ MasterCard ☐ American Express

Charge-card number _____

Expiration date _____

Month _____ Year _____

Prices valid through 12/31/93
693 MICRO

Charge orders also taken by phone:
(714) 821-8380 8 a.m. to 5 p.m. Pacific time
Circulation Dept.
10662 Los Vaqueros Circ., PO Box 3014
Los Alamitos, CA 90720-1264

Editorial comments

I liked: _____

I disliked: _____

I would like to see: _____

Reviewers Needed. If interested, send professional data to Dante Del Corso, Dipartimento di Elettronica, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129, Torino, Italy.

For reader service inquiries, see other side.



PLACE
POSTAGE
HERE

PO Box is for reader service cards only.

IEEE Micro

PO BOX 16508
NORTH HOLLYWOOD CA 91615-6508
USA



Editorial comments

I liked: _____

I disliked: _____

I would like to see: _____

Reviewers Needed. If interested, send professional data to Dante Del Corso, Dipartimento di Elettronica, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129, Torino, Italy.

For reader service inquiries, see other side.



PLACE
POSTAGE
HERE

PO Box is for reader service cards only.

IEEE Micro

PO BOX 16508
NORTH HOLLYWOOD CA 91615-6508
USA



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 38 LOS ALAMITOS, CA

POSTAGE WILL BE PAID BY ADDRESSEE

IEEE COMPUTER SOCIETY

PO BOX 3014
LOS ALAMITOS CA 90720-9804
USA



V.35/36, V.10, V.11, and RS-449. Both 64 and 64M ParaScopes connect to ISDN lines via company interface pods. *Frederick Engineering; \$1,695.*

Reader Service No. 39

No extra power needed for converter

Model 263, an RS-232/RS-422 interface converter, operates under power from the signals applied to the RS-232 interface, provides full-duplex operations for 19.2-Kbps transmit and receive data signals. The 2x2.75x0.75-inch RS-422 signal interface simultaneously serves both screw terminals and an RJ-11 connector. For DTE or DCE configurations, users activate a switch that reverses pins 2 and 3 of the RS-232 connector. *Telebyte Technology; \$89, quantity discounts available.*

Reader Service No. 40

Windows IRMAs announced

IWW 2.1 and IWD 2.0 PC-to-mainframe software includes support for TN3270 over TCP/IP and NetWare for SAA plus enhanced productivity features. IRMA Workstation for Windows 2.1 includes a Quickbar 3270 feature for easy access to Windows and 3270 functions such as session activate/deactivate, file transfer, and a graphical keyboard editor. IRMA Workstation for DOS 2.0 also provides the editor with remote diagnostic support for company-customer interaction. *DCA; \$495 (IWW 2.1), \$425 (IWD 2.0).*

Reader Service No. 41

NS/DOS product restores files

Upstream/PC 2.1.0 provides IBM's Networking Services/DOS with unattended backup and restoration of critical PC and LAN data to and from an MVS mainframe. Operating in both DOS and Windows environments, NS/DOS lets workstations use Advanced Peer-to-Peer Networking technology, while low-memory workstations on the network use APPC LU 6.2 services.

Upstream also handles disaster recovery, addresses issues of client/

server or distributed applications, and lets data in VSAM clusters be archived to tape for long-term storage. *Enterprise Data.*

Reader Service No. 42

T1/E1 chips send digital data

Two T1/E1 line interface unit chips can be used in PCM multiplexing, switching, and transmission systems. Designated the VP14335 and VP14574, the 28-pin PLCC and DIP chips offer a single-chip solution for synthesizing DSX-1 and CCITT G.703 pulses. The VP14335 provides jitter attenuation on the transmitting side of the signal while the VP14574 provides the same on the receiving side. *VLSI Technology; \$9.70 (10,000s).*

Reader Service No. 43

Server runs on nine platforms

Support for Univel's Unixware, Data General Aviiion, Interactive, HP/US, and Acer/Altos platforms has been added to a QX15 ASCII/ANSI terminal that runs X Windows applications. The 68000-based text terminal with GUI and mouse port already supports SCO Unix and Open Desktop, RS/6000 AIX, and Sun environments. To prevent screen flicker, the QX15 provides a 78-Hz refresh rate on its 14-inch, flat, white/green/amber phosphor CRT. *Qume Peripherals; \$699; X server software available for \$200 one-time site license.*

Reader Service No. 44

Modem set corrects errors

Manufacturers can build a complete modem with full data integrity and enhanced features in an area smaller than one half the size of a credit card with the two-chip CL-MD9624EC2. This data/fax/voice modem device set provides MNP4 and V.42 protocols for error correction and MNP5 and V.42bis protocols for data compression at 2,400-bps transfer rates and 9,600-bps facsimile transfers. On-chip communications firmware eliminates the need for software development and debugging. *Cirrus Logic; \$23 (10,000s).*

Reader Service No. 45

VME64 adapters, WAN controller announced

The PT-VME600 FDDI and PT-VME430/432 SCSI-2 adapters join the PT-VME340 VMEbus controller in aiding network communications.

The PT-VME600 Fiber Distributed Data Interface node adapter for VME64 systems makes use of National Semiconductor's two-chip set to provide a 100-Mbps data transmission path between nodes. A dual-attached version of the PT-VME600 adapter supports front-end network applications such as image processing, multimedia, and CAD/CAM. It can also serve as a backbone network to tie together the main components of a distributed system.

The PT-VME430/432 SCSI-2 host adapters promise 17-/19-Mbyte/s sustained data transfer rates between the host and storage peripheral devices. Recommended for Unix applications, these adapters achieve 2-μs/data block overhead and support 8-/16-bit SCSI bus widths, with either differential or single-ended operation.

The PT-VME340 four-port WAN controller is a VMEbus Serial I/O module designed for T1/E1++ rates of 10 Mbps. Built around the Zilog 16C32 controller, this interface is based on many of the features and functions of the Z8530 serial communications controller found in current applications. *Performance Technologies; \$3,996 (PT-VME600, 100s), \$2,570 (/432), \$2,236 (/340, 100s).*

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 189 Medium 190 High 191

Product Summary

Joe Hootman

University of North Dakota

Manufacturer	Model	Comments	R.S.#
Chips			
National Semiconductor	DP84910VHG read channel	Third-generation integrated read channel for hard-disk drives provides 50-Mbps performance and advanced power management. A 5V power supply drives the PQFP chip. <i>\$25 (33 Mbps); \$30 (50 Mbps) 1,000s.</i>	80
Philips Semiconductors	TEA1093 telephone set	IC implements a line-powered, "hands-free" telephone set with on-chip supply regulation, microphone and loudspeaker amplifiers, duplex controller with speech and background noise envelope monitors, and channel-switching logic. The 28-pin surface-mount or DIP chip also works with AC-powered equipment. <i>Hfl 6.00, depending on importing country (10,000s).</i>	81
Systems			
GammaLink	Isofax 400 gateway	Platform integrates GammaNet fax server software and board to provide inbound and outbound faxing capabilities for X.400 network users. The 9,600-bps system converts on-board text and graphics and provides zero fill for high throughput. Multiple boards can be installed in one PC chassis with multiple sending/receiving lines. <i>\$3,450.</i>	82
Philips Semiconductors	CDT610/611 CD systems	With a 3-beam CD drive and disc-loading mechanism, plus digital, analog, keyboard, and display electronics, these systems let users design a CD player for use in Midi hi-fi units, in-car entertainment systems, and portable CD-radio cassette units. A mask-programmed microcontroller implements random-play, forward/reverse track searching, and remote control features.	83
Sparcom	Smart Dock connectors	Intelligent docking stations for 512-Kbyte and 1-Mbyte HP 95LXs connect the palmtop with facsimile machines, electronic information services, printers, desktop PCs, and Macintosh computers. Each system includes Data Exchange software for PC or Mac. <i>From \$169.95.</i>	84
Transtech Parallel Systems	TTM200 memory interface	Module incorporates a 50-MHz i860XP vector processor to provide 400-Mbyte/s sustained data rate and 20-Mbyte memory. TTM200 application development tools, C and Fortran 77 compilers, and symbolic debugger included.	85
VideoLabs	FlexCam camera	A 1/3-in. color CCD camera with two directional stereo microphones mounted on an 18-in. gooseneck arm outputs NTSC video and line-level audio. The integrated system is compatible with most Macintosh and Video for Windows video digitizing boards. The unit's base houses all electronics. <i>\$595.</i>	86

Manufacturer	Model	Comments	R.S.#
Software			
Andersen Consulting	Windows Client Option, V. 1.2	Foundation Cooperative Processing client/server application lets users create enterprisewide systems for Windows 3.1 and OS/2 Presentation Manager environments that incorporate Windows-based personal computers. At generation time, users can select the Windows radio button to generate Windows clients, and can provide the application with PM and Windows user interfaces.	87
Eyring	Pxrom operating system	Modular, real-time porting environment for Motorola IDP boards based on the M68000 supports the Microtec ANSI C compiler on DEC, HP, PC, and Sun hosts for robotics, industrial controls, and data acquisition applications. Developers can configure PDOS modules for various system architectures and acquire runtime licensing only for the used modules. <i>From \$6,000 plus \$175 per license.</i>	88
Intelligent Systems International	Virtuoso programmer	The Virtual Single Processor Programming system based on the company's API-compatible RTXC/MP real-time kernel can be considered as a microkernel while the lowest layers use nano-kernel technology. Runs on 68HC11, 680X0, 96002, 80X86, T2/T4/T8XX, R3000, and TMS320C30/31/40 systems. <i>\$3,995 (single-processor version); \$12,995 (multiprocessors); site developer's license.</i>	89
Mercury Interactive	WinRunner tester	X-Windows automated tester uses context-sensitive technology to ensure test accuracy and reliability and enable test portability between platforms. DECstation, Sparcstation, HP 9000/700, and RS6000 tool interprets high-level, context-sensitive commands and executes them through the GUI as keystrokes and mouse movements. <i>\$6,000 per license.</i>	90
National Instruments	PID control packages	Software accesses PID control through graphical or text-based programming tools for Labview for Windows and LabWindows for DOS. PC and Macintosh packages offer P, PI, PD, and PID algorithms; lead/lag compensation; automatic/manual control mode; and error-squared PID. <i>\$295.</i>	91
Oasys	68K Cross Tool Kit	Development tools for Alpha AXPs include Green Hills C++, C, Fortran, and Pascal cross compilers; the Oasys 68K Cross Assembler/Linker system; and Multi, a window-oriented, source-level debugger. DECstation and Vax systems under Open VMS or Ultrix, most Unix workstations, and PCs running MS Windows can also use the tools.	92
Miscellaneous			
Tadpole Technology	Sparcbook keyboards	European local language keyboards support Sparcbook notebook workstations, enhancing their international Sun Type-4 and Type-5 keyboards. Each integral keyboard includes a mouse key, 12 function keys, and 82 full-size keys.	93

Information for Authors

April 1993

Who we are

IEEE Micro, a bimonthly publication of the IEEE Computer Society, reaches an international audience of microcomputer and microprocessor designers, system integrators, and users. Readers seek to increase their technical knowledge of computers and peripherals; systems, components, and sub-assemblies; communications, instrumentation, and control equipment; and software.

What we publish

IEEE Micro publishes original works about 5,500 words long (about 20 double-spaced typed pages that include explanatory figures, tables, and programs). These works discuss the design, performance, or application of microcomputer and microprocessor systems. Readers welcome tutorial material, review papers, and discussions of standards. Topic areas include

- systems
- fault tolerance
- languages
- application software
- algorithms
- hardware/software design and implementation
- architecture
- data acquisition
- operating systems
- artificial intelligence
- communications

Submitting your manuscript

Submit six copies of your manuscript and a 50- to 70-word abstract with keywords, your mailing address, phone and fax numbers, and electronic mail address directly to:

Dante Del Corso
Editor in Chief, *IEEE Micro*
Dipartimento di Elettronica
Politecnico di Torino
C.so Duca degli Abruzzi, 24
10129 Torino, Italy
Telephone: + 39 11 564 4044; fax: + 39 11 564 4099
Comppmail: d.delcorso; Bitnet: delcorso@itopoli;
Internet: delcorso@polito.it

or

Maurice Yunik
Associate Editor in Chief
Dept. of Electrical Engineering
University of Manitoba
Winnipeg, Manitoba R3T 2N2 Canada
Telephone: (204) 474-8517; fax: (204) 275-0261
Internet: yunik@eeserv.ee.umanitoba.ca

All manuscripts pass through a peer-review process consistent with other professional-level technical publications. This process may take three months, and referees may require revisions to parts of your work. If a manuscript exceeds the specified length, it will be shortened.

Successful contributions avoid the style of transactions and academic journals. They sufficiently introduce the material, place it in context with similar works, describe the practical or potential applications of the material presented, and discuss both pros and cons of the approach. At least 20 percent of the article should be tutorial in nature. Brief literature surveys do not satisfy these requirements.

Upon accepting your manuscript for publication, the Editor in Chief will ask you to supply three copies of any revised draft, plus drawings, photographs, equations, and programs; an electronic version; and biographies and photos of all authors. In addition, you must sign a release transferring copyright to the IEEE (excepting certain key rights retained by the author). Details follow under the Copyright heading.

Submit the hard copies, including illustrations and references or bibliographies, printed on one side only of 8 1/2 × 11-inch paper and double spaced with at least 1 1/2-inch margins. Send an electronic copy on floppy disk or via Compmail or Internet. All electronic files should retain any text-formatting codes you use and identify the formatter used. Refer to the Computer Society's Electronic Submittal Guide for further details. Disks must be Macintosh-compatible or 5.25-inch, IBM PC-compatible, and running DOS Version 2.10 or newer.

For further guidance, contact:

Marie English, Managing Editor, *IEEE Micro*
10662 Los Vaqueros Circle; PO Box 3014
Los Alamitos, CA 90720-1264
Telephone: (714) 821-8380; fax: (714) 821-4010
Internet: m.e.english@compmail.com

Professional editors on the *IEEE Micro* staff thoroughly edit accepted manuscripts. This collaborative process between author and editor results in a concise, well-worded article. Editing covers grammar and punctuation; content (flow, meaning, clarity, directness, organization); and style (conformance to house style).

Copyright

When parts of a manuscript have already been published elsewhere, the author must seek permission from the original publisher. The article will acknowledge the permission; for example, "Section ... and figure ... appeared in They are reprinted with permission of the publisher." If IEEE originally published this material, permission is automatically granted, but the original publication must still be cited. Detailed descriptions of author and IEEE rights appear on the IEEE copyright form (yellow sheet).

Manuscript date of receipt

We will publish the date we received the manuscript; just request this when submitting the final version of an accepted article.

Writing tips

Readers welcome clear, accurate articles presented in logical sequence. Let readers know in the first paragraph why your subject is important; give them a reason to continue reading. Define your problem and discuss your solution and any trade-offs. Augment your discussion with examples, tables, diagrams, charts, and photographs to help readers grasp your point. End by putting your topic in perspective and stating any future work plans. Remember, all readers won't be familiar with your specialty; you will have to explain unusual terms or intricate processes.

Readers move swiftly through articles written in the active voice and containing short words, short sentences, and concrete examples. (An active voice example: "This scheme contains two main buses" NOT "Two main buses are contained in this scheme.") Avoid jargon, explain acronyms, and simplify your language. For example, use "to" NOT "for the purpose of" and use "can" NOT "has the capability to." In other words, write the way you talk.

As you can see, magazine style differs from journal and report styles.

References and bibliographies

References substantiate points made in the text or direct readers to other points of view or important works. Do not overdo it, however; most articles need less than 10 citations. They appear in numerical order in the article and in a separate section at the end of the article. Citations in the text appear as Arabic superscripts, for example, Smith.¹ (Use square brackets if your word processor does not allow superscripts.)

Cited sources should be available to the reader; don't include unpublished works. Any abbreviations should follow *IEEE Micro* usage; see a recent issue for examples. When in doubt, spell it out.

You should attempt to provide full bibliographic data as a courtesy to your readers. A complete citation includes author(s); title of article or chapter; title of journal, book, proceedings, or dissertation; publisher's name, city, and state for books and dissertations; complete address for private technical reports; year published; and inclusive page numbers.

Illustrations

Submit photocopies of artwork, rather than originals, for the initial manuscript review. All final illustrations and drawings should be clear and submitted in hard copy (on separate sheets). *IEEE Micro* reproduces your original halftones, machine-made graphs, computer printouts, and electronically produced artwork. Artists will redraw all other art to meet house standards. Photographic prints should have good contrast and be at least 7.5 × 12 cm (3 × 5 inches). Check to see that all artwork is accurate and unambiguous, and uses the same terms as the text. Number, caption, and cite in text all illustrations and tables. Captions are short, for example:

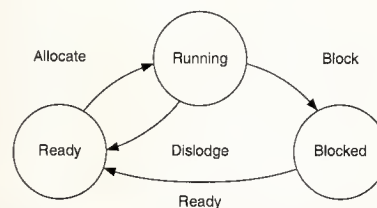


Figure 1. Task states and transitions. (Copyright 1995 William Jones. Reprinted by permission.)

Biographical sketch and photograph

Submit a photograph and biographical sketch of each author. Good-quality, black-and-white glossy photographs with good contrast, preferably 7.5 × 12 cm (3 × 5 inches) in size, reproduce best. Limit biographical sketches to 75 words and include, in the following order: current positions and technical interests, prior professional experience and other important activities, education, professional affiliations, and current address. See a recent issue of *IEEE Micro* for examples.

Micro News

continued from p. 8

and Motorola, Inc. entered a joint venture to develop wireless electronics technology for remote and automated meter reading (RAMR). The new products will replace today's on-site visual inspections and hand-held data collection terminals.

The joint venture will take the form of an Atlanta, Georgia, design center that will also provide integrated solutions for water, gas, heat, and electricity utility meters on a global basis. Motorola will manufacture products, while Schlumberger provides metering products, marketing, sales, and customer services to utilities worldwide. Both companies will be equal owners of the company with equal representation on its board.

Electronic pen clipboards. PI Systems Corporation, developer of Infolio electronic pen clipboards, and Business Partner Solutions Inc., developer of AS/Messenger RadioPac wireless communications software, agreed to form a cooperative marketing partnership. The companies plan to develop wireless electronic pen clipboard services especially for health care providers, using Motorola InfoTAC and GE Ericsson Mobidem modems.

Designed to speed up the flow of error-free information to an organization's database, the real-time, pen-based interaction will supply database information to clinicians, whether in or out of the hospital environment. Expected benefits include reduced paperwork and billing cycles, real-time access to extensive patient information, and natural pen input for professionals.

Video/audio compression. Texas Instruments and C-Cube Microsystems recently announced an agreement to develop video and audio compression products. These products will be used in digital cable television and Direct Broadcast Satellite TV, HDTV, compact disc-based consumer video, and per-

sonal computer multimedia applications.

The agreement includes technology and product development exchanges between the two companies. It also provides each company with rights to develop derivative products of the other company's current and future MPEG and JPEG coder/decoder products. C-Cube will receive access to TI's advanced CMOS process technologies and production facilities. Specific applications for these products include Compact Disc-Interactive, CD-based karaoke players, CD-based video games, and new digital TV broadcast receivers.

Bill O'Meara of C-Cube stated, "With TI's marketing and production strength behind MPEG and JPEG, the digital video market is poised for explosive growth." Walden C. Rhines of TI's Semiconductor Group said, "The application of DSP techniques to consumer products promises to be a tremendous growth market for the semiconductor industry in the 1990s."

Traffic control. A neural network computer program originally developed to help military pilots deal with enemy threats may soon be used to ease traffic congestion, according to computer scientists at the Georgia Tech Research Institute in Atlanta. The TERMINUS traffic control program senses traffic conditions and regulates the operation of signal lights to optimize the flow of vehicles.

TERMINUS, short for Traffic Event Response and Management for Intelligent Navigation Utilizing Signals, runs on Sun Sparc and similar workstations. It represents each intersection as a neuron and each street segment between intersections as a neural interconnection.

The program displays an animated color map of streets, parking lots and the number of cars in them, and traffic conditions in potential problem areas. TERMINUS even provides a computer-generated sound of crashing vehicles to alert its operators to traffic accidents.

The initial application for the system simulated traffic conditions at the Atlanta Braves stadium to demonstrate

Micro bits

- Georgia Tech (xspice@gtlgatech.edu) offers its **XSPICE simulator** through a no-cost license agreement and a \$200 distribution charge. Useful when mixing system and analog simulations, the 1992 Unix SPICE extension in source code form is compatible with the original code.

- Dial 1-900-680-DEAL 24 hours/day, 7 days/week for **discount prices** on microcomputers, peripherals, printers, and fax machines. The P.C. Discount Shopper supplies biweekly updated product information available to callers via phone or fax. Each call costs \$1.95/minute.

- A *Catalog of National ISDN Solutions for Selected NIUF Applications* selling for \$44.50 describes 30 **ISDN applications** and the required equipment and services for building the applications. Contact the National Technical Information Service at (703) 487-4650 to order; specify PB 93-162881.

- X Business Group, a market research company, reports that the value of all **X specific products and services** sold worldwide increased 60 percent during 1992 to top \$800 million. It also sets the worldwide installed base of X capable seats at over 2 million.

- The IEEE Computer Society has donated "**instant libraries**" to engineers and scientists at 30 sites in Eastern Europe and the former Soviet Union. Each library valued at \$15,000 contains 200 authored books, reprint collections, and conference proceedings.

how signal light settings might be coordinated during special events. The next stage will create a hardware installation that can be integrated into an overall traffic management control system. Researchers will then join the system to a central traffic control computer and determine how it can accept and process data inputs from the sensors.

Larger applications will require the integration of complex geographic information systems into the work of TERMINUS.

Applications sought for manufacturing fellowships

US Commerce Secretary Ronald H. Brown announces the start-up of a program to place US engineers in Japanese manufacturing firms for up to one year. The goals of the Manufacturing Technology Fellowship project are to help US engineers learn more about—and then use—Japanese manufacturing practices and to promote long-term professional exchanges.

The program is accepting applications from US engineers sponsored by their companies. Fellowships will last about 15 months and include three months of intensive Japanese language and culture training in the US. Participants will learn about Kanban, just-in-time manufacturing, total quality control, and other techniques. Sixty Japanese companies will act as host organizations.

Potential candidates should contact project representatives at the US Department of Commerce by fax at (202) 482-4826. Applications must arrive by July 16, 1993.

VLSI Design 93 meets in Bombay

The Sixth International Conference on VLSI Design met in Bombay, India, January 3-6, with over 400 attendees from around the world. The conference was organized in cooperation with the ACM Special Interest Group on Design Automation, the IEEE Computer Society's Technical Committees on

Design Automation and VLSI, and the IEEE Circuits and Systems Society. The conference also received support from the Department of Electronics of the Government of India.

Centering on chips, boards, and systems in the 90s, the conference began with the keynote address of Osamu Karatsu of NTT LSI Labs, Japan, on the "History and Future Directions of VLSI CAD and Design: A Japanese Perspective."

The two-day technical program consisted of 70 papers and 9 posters selected from a total of 186 submissions. Topics included logic synthesis, design for testability, physical design, testing, high-level synthesis, VLSI algorithms and architectures, parallel CAD, CAD frameworks, logic design, circuit design, and delay fault testing.

The conference also organized five one-day tutorials on topics such as FPGAs and DSP, exhibits of Indian CAD/CAE systems and VLSI/PCB design services, and two design contests for the Indian participants.

Next year's meeting will be held January 5-8 in Calcutta (see Call for Papers in March 1993 *Computer*). For information, contact Rochit Rajsuman (rajsuman@alpha.ces.cwru.edu).

PDA interest explored

According to a BIS Strategic Decisions survey, one third of the respondents indicated that they would buy a personal digital assistant, even though PDAs are not yet on the market. Surprisingly, price was not a key issue for potential buyers.

The PDA concept involves stylus-based computing, handwriting recognition, electronic organizing, word processing, spreadsheets, database management, and other wireless communications functions.

Although the PDA market is often described as the future of pen computing, the survey showed that less than 25 percent of respondents prefer pen input; 54 percent preferred the keyboard.

Sales managers proved to be the best

target market for PDAs. They are open to and interested in the concept and have the decision-making power to implement PDA use in their departments.

BIS Strategic Decisions, an international organization of industry analysts, produces a series of research reports on PDAs called *Emerging Markets for Personal Digital Assistants*. Interested parties can obtain pricing information from Robin Osborne, phone (617) 982-9500 or fax (617) 878-6650.

Literature

PC Design Guide includes a schematic summary sheet of chip sets available to designers of PC compatibles, and a list of manufacturers and phone numbers. *Annabooks, 15010 Avenue of Science, #101, San Diego, CA 92128-3421; (619) 673-0870; fax (619) 673-1432; \$139.*

Engineers can now obtain the fourth edition of a 220-page handbook on making low-level measurements. Featured are step-by-step procedures, instructions, and a glossary of terms. *Keithley Instruments, Inc., 28775 Aurora Road, Cleveland, OH 44139; phone (216) 248-0400; fax (216) 248-6168; free.*

Need a how-to pocket guide to help you establish an open systems environment within your company? *A Very Useful Guide—Buying Open Systems* will help with the terminology and issues associated with purchasing open systems and the general process. *The 88open Consortium Ltd., 100 Homeland Court, Suite 800, San Jose, CA 95112; phone (408) 436-6600; fax (408) 436-0725; \$8.*

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 198 Medium 199 High 200

FOR DISPLAY ADVERTISING INFORMATION, CONTACT:

Southern California and Mountain States: Richard C. Faust, Douglas C. Faust, 24050 Madison Street, Suite 101, Torrance, California 90505; Tel: (310) 373-9604; Fax: (310) 373-8760.

Northern California and Pacific NW: William W. Hague, 9017 Peacock Hill Road, Gig Harbor, Washington 98332; Tel: (206) 858-7575; Fax (206) 858-7576.

East Coast: Gail A. Frank, Nancy Inserra, 82 Bethany Road, Suite 4, Hazlet, New Jersey 07730; Tel: (908) 264-1100; Fax: (908) 264-4340.

New England: Paul Gillespie, PO Box 6444, Holliston, Massachusetts 01746; Tel: (508) 429-8907; Fax (508) 429-8684.

Southwest: Frank E. Johnson, 3601 Smith-Barry Road, Suite 103, Arlington, Texas 76013; Tel: (817) 275-2651; Metro Voice/Fax: (817) 265-3811.

Advertising Manager: Heidi Rex, 10662 Los Vaqueros Circle, Los Alamitos, California 90720-1264; Tel: (714) 821-8380; Fax: (714) 821-4010.

For production information, conference, and classified advertising, contact Marian Tibayan. *IEEE MICRO*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, California 90720-1264; Tel: (714) 821-8380; Fax: (714) 821-4010; email: m.tibayan@compmail.com.

RS # Page #

Actel	12	93
Andersen Consulting	87	99
AST Research	38	96
Cirrus Logic	45	97
Covox	22	94
Crystal Semiconductor	23	95
DCA	41	97
Digital Communications Associates	35	96
Enterprise Data	42	97
Exemplar Logic	10	93
Eyring	88	99
Frederick Engineering	39	97
GammaLink	82	98
General DataComm	34	96
HEM Data	26	95
Hot Interconnects '93	—	2-3
Hyperception	20	94
Integrated Systems Technologies	19	94
Intelligent Systems International	89	99
Japan Computer Corp.	31	96
Linear Technology	24	95
Link Technologies	33	96
Lucas Deeco	30	95
Mentor Graphics	16	94
Mercury Interactive	90	99
MicroSim	13	93
National Instruments	91	99
National Semiconductor	80	98
NeoCAD	11	93
Network Software Associates	32	96
Oasys	92	99
Perception Technology	28	95
Performance Technologies	46	97
Philips Semiconductors	81, 83	98
Qume Peripherals	44	97
RGB Spectrum	29	95
Schlumberger Technologies	18	94
Sparcom	84	98
Spectrum Signal Processing	27	95
Standard Microsystems	17	94
Tadpole Technology	93	99
Telebyte Technology	40	97
Texas Instruments	21	94
Transtech Parallel Systems	85	98
VideoLabs	86	98
VLSI Technology	37, 43	96, 97
White Pine	36	96
Wind River Systems	15	94
World Precision Instruments	25	95
Xilinx	14	94

Moving?

**PLEASE NOTIFY
US FOUR WEEKS
IN ADVANCE**

Name (Please Print)

New Address

City

State/Country

Zip

Mail to:
IEEE Computer Society
Circulation Department
PO Box 3014
10662 Los Vaqueros Circle
Los Alamitos, CA 90720-1264

**ATTACH
LABEL
HERE**

- This notice of address change will apply to all IEEE publications to which you subscribe.
- List new address above.
- If you have a question about your subscription, place label here and clip this form to your letter.

THE FOLLOWING INFORMATION IS AVAILABLE:

Contact the Publications Office;
to facilitate handling, please request by number.

- Membership application, student #203, others #202
- Publications catalog #201
- Compmail electronic mail brochure #194
- Technical committee list/application #197
- Chapters lists, start-up procedures #193
- Student scholarship information #192
- Volunteer leaders/staff directory #196
- IEEE senior member grade application #204
(requires ten years practice and significant performance in five of those ten)

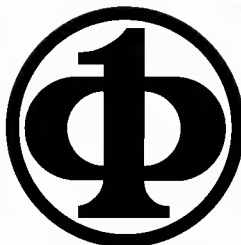
To check membership status or report a change of address, call the IEEE toll-free number, 1-800-678-4333. Direct all other Computer Society-related questions to the Publications Office.

PURPOSE

The IEEE Computer Society advances the theory and practice of computer science and engineering, promotes the exchange of technical information among 100,000 members worldwide, and provides a wide range of services to members and nonmembers.

MEMBERSHIP

Members receive the acclaimed monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.



IEEE COMPUTER SOCIETY

A member society of the
Institute of Electrical and Electronics Engineers, Inc.

PUBLICATIONS AND ACTIVITIES

Computer. An authoritative, easy-to-read magazine containing tutorial and in-depth articles on topics across the computer field, plus news, conferences, calendar, interviews, and product reviews.

Periodicals. The society publishes eight magazines and seven research transactions. Refer to membership application or request information as noted at left.

Conference Proceedings, Tutorial Texts, Standards Documents. The Computer Society Press publishes more than 100 titles every year.

Standards Working Groups. More than 100 of these groups produce IEEE standards used throughout the industrial world.

Technical Committees. More than 30 TCs publish newsletters, provide interaction with peers in specialty areas, and directly influence standards, conferences, and education.

Conferences/Education. The society holds about 100 conferences each year and sponsors many educational activities, including computing science accreditation.

Chapters. Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

OMBUDSMAN

Members experiencing problems — magazine delivery, membership status, or unresolved complaints — may write to the ombudsman at the Publications Office.

EXECUTIVE COMMITTEE

President: James H. Aylor*
University of Virginia
Thornton Hall/Electrical Engineering
Charlottesville, VA 22903
Phone: (804) 924-6100
Fax: (804) 924-8818
E-mail: jha@virginia.edu

President-Elect: Laurel V. Kaleda*
Past President: Bruce D. Shriver*

VP, Technical Activities: Joseph Boykin (1st VP)*
VP, Conferences and Tutorials: Anneliese Von Mayrhauser (2nd VP)*
VP, Educational Activities: Gerald L. Engel†
VP, Membership Activities: Fiorenza C. Albert-Howard*
VP, Press Activities: Ronald G. Hoelzeman†
VP, Publications: Barry W. Johnson†
VP, Standards Activities: Gary S. Robinson†

Secretary: Mario R. Barbacci*
Treasurer: Michael Evangelist†
IEEE Division V Director: Bill D. Carroll†
IEEE Division VIII Director: V. Tom Rhyne*
Executive Director: T. Michael Elliott†

*voting member of the Board of Governors
†nonvoting member of the Board of Governors

BOARD OF GOVERNORS

Term Expiring 1993:

Fiorenza C. Albert-Howard, Jon T. Butler,
Michael C. Mulder, Yale N. Patt, Benjamin W. Wah,
Ronald Waxman, Akihiko Yamada

Term Expiring 1994:

Mario R. Barbacci, L. Felipe Cabrera, Wolfgang K. Giloi, Guyline M.
Pollock, John P. Riganati, Ronald D. Williams, Thomas W. Williams

Term Expiring 1995:

Fletcher J. Buckley, Doris L. Carver, Elliot J. Chikofsky,
Joanne E. DeGroat, Michael J. Flynn,
Mary Jane Irwin, Grace C.N. Wei

Next Board Meeting

November 21, 1993, 8:30 a.m.
Santa Clara Marriott, Santa Clara, Calif.

SENIOR STAFF

Executive Director: T. Michael Elliott
Publisher: H. True Seaborn
Director, Conferences and Tutorials: Anne Marie Kelly
Director, Finance and Information Services: Deborah Rafal
Director, Board and Administrative Services: Violet S. Doan
Assistant to the Executive Director: Sandra K. Pfau

COMPUTER SOCIETY OFFICES

Headquarters Office

1730 Massachusetts Ave. NW
Washington, DC 20036-1903
Phone: (202) 371-0101
Fax: (202) 728-9614
E-mail: hq.ofc@compmail.com

Publications Office

10662 Los Vaqueros Cir.
PO Box 3014
Los Alamitos, CA 90720-1264
Membership and General Information:
(714) 8218380
membership@compmail.com
Publication Orders: (800) 272-6657
Fax: (714) 821-4010
E-mail: cs.books@compmail.com

European Office

13, Ave. de L'Aquilon
B-1200 Brussels, Belgium
Phone: 32 (2) 770-21-98
Fax: 32 (2) 770-85-05
E-mail: euro.ofc@compmail.com

Asia/Pacific Office

Doshima Building
2-19-1 Minami-Aoyama, Minato-ku
Tokyo 107, Japan
Phone: 81 (3) 3408-3118
Fax: 81 (3) 3408-3553
E-mail: tokyo.ofc@compmail.com



IEEE OFFICERS

President: Martha Sloan
President-Elect: H. Troy Nagle
Past President: Merrill W. Buckley, Jr.

Secretary: Souguil J.M. Ann
Treasurer: Theodore W. Hissey, Jr.

VP, Educational Activities: Edward A. Parrish
VP, Professional Activities: Charles K. Aledxander
VP, Publication Activities: Helen M. Wood

VP, Regional Activities: Luis T. Gandia
VP, Standards Activities: Wallace S. Read
VP, Technical Activities: Donald M. Bolle

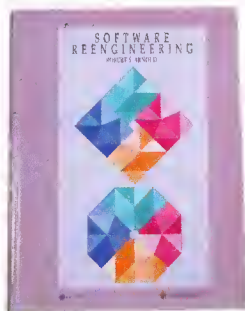
IEEE COMPUTER SOCIETY PRESS

SOFTWARE REENGINEERING

edited by Robert S. Arnold

This tutorial covers a wide variety of interesting software reengineering approaches and introduces its themes, strategies, technology, and risks. Explores and defines software reengineering concepts and processes, tools and techniques, capabilities and limitations, risks and benefits, and research possibilities.

Sections: Context and Definition, Business Process Reengineering, Strategies and Economics, Reengineering Experience and Evaluation, Technology for Reengineering, Data Reengineering and Migration, Source Code Analysis, Software Restructuring and Translation, Documenting Existing Programs, Reengineering for Reuse, Reverse Engineering and Design Recovery, Object Recovery, Knowledge-Based Program Analysis.



688 pages. ISBN 0-8186-3272-0. April 1993. Hardcover.
Catalog No. 3272-01 — List Price \$79.00 Members \$64.00

GROUPWARE:

Software for Computer-Supported Cooperative Work

edited by David Marca and Geoffrey Bock

This book focuses on the development of new software that enhances cooperation and augments human capability. It is a collection of distinctions, approaches, methods, and examples that are transforming the development of computer systems for groups. Describes key topics such as user interface technologies, ways to design software to enhance human capabilities, methods and applications to coordinate human activity, and the effectiveness of groupware. The text also concentrates on the design of software to fit the way groups interact in specific work situations.

Sections: Introduction, Groups and Groupware, Conceptual Frameworks, Design Methods, Enabling Technologies - System-Related, Enabling Technologies

- UI-Related, Computer-Supported Meetings, Bridging Time and Space, Coordinators, What Makes Systems Effective, Bibliography, Index.



592 pages. ISBN 0-8186-2637-2.
August 1992. Hardcover.
Catalog No. 2637-01
List Price \$75.00
Members \$45.00



IEEE COMPUTER SOCIETY
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

Call toll-free 1-800-CS-BOOKS
in CA (714) 821-8380
or FAX (714) 821-4641

SOFTWARE MANAGEMENT, 4th Edition

edited by Donald J. Reifer

The tutorial includes both original material and reprints that amplify related management theories, concepts, tools, and techniques and provide guidelines to improve the practice. Its text also includes coverage of process

assessment, metrics, and risk management topics. Provides managers with an in-depth study of the available technology for managing the process, product, and people involved in software development and maintenance.



Sections: Software Process, Project Management, Planning Fundamentals, Organizing for Success, Staffing Essentials, Direction Advice, Visibility and Control, Risk Management, Metrics and Measurement, Software Engineering Technology Transfer, Support Material.

664 pages. ISBN 0-8186-3342-5. April 1993. Hardcover.
Catalog No. 3342-01 — List Price \$79.00 Members \$64.00

COMPUTER-AIDED SOFTWARE ENGINEERING (CASE), 2nd Edition

edited by Elliot Chikofsky

This new edition presents new information on CASE technology, and examines its present state, how its concepts have fared over time, and how it looks as a technology for the future. Features more than 35% new papers and combines the latest papers in the field with background articles that allow the reader to see how the field is evolving.



Sections: CASE Environments and Tools: Overview, Role of Assistants and Expert System Technology in CASE, Evolution of Software Development Environment Concepts, Role of Prototyping in CASE, Role of Data Browsing Technology in CASE, Tailoring Environments (Extension, Meta-Specification, Generation), Issues of Evaluating Tools and Managing CASE.

184 pages. ISBN 0-8186-3590-8.
January 1993. Softcover.
Catalog No. 3590-05
List Price \$35.00
Members \$25.00

COMING SOON

SIMULATION VALIDATION

by Peter L. Knepell and Deborah C. Arango
Catalog # 3512-04

SOFTWARE ENGINEERING: A European Perspective
edited by Richard H. Thayer and Andrew D. McGettrick
Catalog # 2117-01

Readings in REAL-TIME SYSTEMS
edited by Y. H. Lee and C. M. Krishna
Catalog # 2997-01

Readings in DISTRIBUTED COMPUTING SYSTEMS
edited by Thomas L. Casavant and Mukesh Singhal
Catalog # 3032-01

IEEE

MICRO

OCTOBER 1993

Chips, Systems, Software, and Applications

A WORLD OF EMBEDDED SYSTEMS:

3 Powerful New Chips From Japan



SPECIAL FEATURES:

- **MOTOROLA POWER PC**
- **END-USER PERFORMANCE, PART 2**
- **DISTRIBUTED SYSTEM PERFORMANCE**



IEEE COMPUTER SOCIETY



THE INSTITUTE OF ELECTRICAL AND
ELECTRONICS ENGINEERS, INC.

IEEE MICRO

Published by the IEEE Computer Society

Volume 13 Number 5

October 1993

F E A T U R E S

6 Guest Editor's Introduction: Toward a World Filled with Computers

Ken Sakamura

12 The Gmicro/500 Superscalar Microprocessor with Branch Buffers

Kunio Uchiyama, Fumio Arakawa, Susumu Narita, Hirokazu Aoki, Ikuya Kawasaki, Shigezumi Matsui, Mitsuyoshi Yamamoto, Norio Nakagawa, and Ikuo Kudo

Featuring RISC-like dual-pipeline structure and resident dedicated branch buffers to achieve a 132-MIPS preprocessing rate

24 The μ VP 64-Bit Vector Coprocessor: A New Implementation of High-Performance Numerical Computation

Makoto Awaga and Hiromasa Takabashi
Integrating 1.5 million transistors on one chip for use in open systems with 206-Mflops performance

37 Fuzzy Inference and a Fuzzy Inference Processor

Kazuo Nakamura, Narumi Sakashita, Yasubiko Nitta, Ken'ichi Shimomura, and Takeshi Tokuda

Operating at 200,000 fuzzy logic inferences per second for high-speed control applications

SPECIAL FEATURES

49 Special Report: HDTV Research in Japan

David K. Kabaner

Looking at some of the more interesting displays at NHK's annual open house

54 The PowerPC 601 Microprocessor

Michael C. Becker, Michael S. Allen, Charles R. Moore, John S. Mubich, and David P. Tuttle

Implementing the PowerPC architecture for use in low-cost desktops to high-performance multiprocessors

69 Spearmints: Hardware Support for Performance Measurements in Distributed Systems

Uwe Kleinbans, Joerg Kaiser, and Karol Czaja

Using a system of sensors for fine-grained measurements while minimally impacting the observed system

79 How Does Processor MHz Relate to End-User Performance? Part 2 of 2

Steven W. White, Phil D. Hester, Jack W. Kemp, and G. Jeanette McWilliams

Supporting the claim that cycle time alone does not determine performance

Cover Image: Francesco Ruggeri, Image Bank West

Cover Design: Design and Direction

Circulation: *IEEE Micro* (ISSN 0272-1732) is published bimonthly by the IEEE Computer Society, PO Box 3014, Los Alamitos, CA 90720-1264; IEEE Computer Society Headquarters, 1730 Massachusetts Ave., NW, Washington, DC 20036-1903; IEEE Headquarters, 345 East 47th St., New York, NY 10017. Annual subscription: \$23 in addition to IEEE Computer Society or any other IEEE society member dues; \$42 for members of other technical organizations. This journal is also available in microfiche form.

Postmaster: Send address changes and undelivered copies to *IEEE Micro*, PO Box 3014, Los Alamitos, CA 90720-1264. Second-class postage is paid at New York, NY, and at additional mailing offices. Canadian GST#125634188.

Copyright and reprint permissions: Abstracting is permitted with credit to the source. Libraries are permitted to photocopy beyond the limits of US Copyright Law for the private use of patrons 1) those post-1977 articles that carry a code at the bottom of the first page, provided the per-copy fee indicated in the code is paid through the Copyright Clearance Center, 27 Congress St., Salem, MA 01970; and 2) pre-1978 articles without fee. For all other copying, reprint, or republication permissions, write to Permissions Editor, *IEEE Micro*, PO Box 3014, Los Alamitos, CA 90720-1264. Copyright © 1993 by the Institute of Electrical and Electronics Engineers, Inc. All rights reserved. Printed in USA.

DEPARTMENTS

- 2 Micro Law**
Protecting industrial property rights
- 4 Micro Review**
Programming Windows
- 90 Software Report**
Cooperation: A new watchword
- 93 New Products**
Communications/displays; DSP components; CAD tools
- 98 Product Summary**

*Reader Service cards, p. 96A/B;
Advertiser/Product Index/
Moving Coupon, p. 104;
Computer Society information,
cover 3*

IEEE Computer Society

PO Box 3014, Los Alamitos, CA 90720-1264
(714) 821-8380; menglish@computer.org

Editorial: Unless otherwise stated, bylined articles and descriptions of products and services reflect the author's or firm's opinion; inclusion in this publication does not necessarily constitute endorsement by the IEEE or the IEEE Computer Society. Send editorial correspondence to *IEEE Micro*, PO Box 3014, Los Alamitos, CA 90720-1264. All submissions are subject to editing for style, clarity, and space considerations.

EDITOR IN CHIEF

Dante Del Corso
*Politecnico di Torino**

ASSOCIATE EDITORS IN CHIEF

K.E. Grosspietsch
GMD

Ashis Khari
Mips Technologies, Inc.

Maurice Yunik
University of Manitoba**

EDITORIAL BOARD

Stephen L. Diamond
Picosoft, Inc.
Joe Hootman
University of North Dakota
Victor K.L. Huang
National University of Singapore
David K. Kahaner
*National Institute of Standards
and Technology*
Hubert D. Kirmann
Asea Brown Boveri Research Center
Priscilla Lu
AT&T
Richard Mateosian
Teresa H. Meng
Stanford University
Nadine E. Miner
Sandia National Laboratories

Gilles Privat
France Telecom
Ken Sakamura
University of Tokyo
John L. Schmalzel
University of Texas at San Antonio
Arun K. Sood
George Mason University
John W. Steadman
University of Wyoming
Richard H. Stern
Oblon, Spivak, McClelland, Maurer & Neustadt
Osamu Tomisawa
Mitsubishi Electric Corporation
Philip Treleven
University College London
Uri Weiser
Intel Israel, Ltd.

MAGAZINE OPERATIONS

COMMITTEE

James J. Farrell, III (chair)
Valdis Berzins
B. Chandrasekaran
Carl Chang
Manuel d'Abreu
Dante Del Corso
John A.N. Lee
Ted Lewis
Michael J. Quinn
Peter R. Wilson

PUBLICATIONS BOARD

Barry W. Johnson (chair)
James J. Farrell, III
Ronald G. Hoelzeman
Mary Jane Irwin
Ming T. (Mike) Liu
Michael C. Mulder
Theo Pavlidis
David C. Rine
Sallie V. Sheppard
Harold Stone
Kishor Trivedi

STAFF

Marie English
Managing Editor
Dick Price
Staff Editor
Anna Taylor
Assistant Editor
H.T. Seaborn
Publisher
Marilyn Pores
Editorial Director, CS Magazines
Jay Simpson
Art Director
Joseph Daigle
Design/Production
John Gill
Membership/Circulation Manager
Heidi Rex
Advertising Manager
Marian Tibayan
Advertising Coordinator

* Submit six copies of all articles and special-issue proposals to Dante Del Corso,
Dipartimento di Elettronica, Politecnico di Torino,
C.so Duca degli Abruzzi, 24, 10129 Torino, Italy; phone +39-11-564-4044;
Comppmail: d.delcorso; Internet: delcorso@polito.it; Bitnet: delcorso@itopoli
or

** Maurice Yunik, Dept. of Electrical Engineering, University of Manitoba,
Winnipeg, Manitoba R3T 2N2, Canada; phone (204) 474-8517; yunik@eeserv.cc.umanitoba.ca

Author guidelines for article submission
are available from the West Coast office (address at left).



Richard H. Stern

Graham & James

2000 M Street, N.W.

Suite 700, Washington,

DC 20036-3307

Protecting industrial property rights

modeling. Technique of system analysis and design using mathematical or physical idealizations of all or a portion of the system. Completeness and reality of the model are dependent on the questions to be answered, the state of knowledge of the system, and its environment.

model. A mathematical or physical representation of system relationships. See: mathematical model.

mathematical model (analog computers). A set of equations used to represent a physical system.

—*IEEE Standard Dictionary of Electrical and Electronics Terms* (3rd ed., 1984)

Legal systems can be represented in terms of models, too. The models are not sets of equations, however. They are a set of descriptions of the system or its characteristics. Thus, the US patent and copyright law systems can be described, as they are here, in terms of models. Representing legal systems in terms of models facilitates understanding of their operation and permits some amount of simulation. That may permit us to ascertain the circumstances that impose stresses on the systems and may strain them beyond their limits. The results of simulations may also suggest modifications of the modeled system, to avoid or lessen the effects of stressful circumstances.

The conventional approach to intellectual property protection in the United States has relied almost entirely on two models—those of patent law and copyright law. These two models of legal protection have limitations, however, particularly in regard to late 20th century com-

puter software technology. Systems based on these models are particularly unsuited to protecting noncode aspects of computer software—to protecting against nonverbatim, nonliteral copying of computer programs. The time may now be ripe for consideration of other legal models for protecting software, at least nonliteral, noncode aspects of software.

The patent model

The patent law model for industrial property rights requires as conditions of legal protection: utility, novelty, and a high level of technical merit or technological advance. That high level of advance is variously called inventive level, inventive step, or nonobviousness. It refers to work substantially above the routine, an advance in technology that a person of ordinary skill in the field would not have conceived and reduced to working form.

To determine whether a product that is a candidate for patent protection meets the conditions for protection, a patent system relies on examination by technical experts before any legal protection becomes effective. This is an expensive and time-consuming procedure, both for the applicant and the government. But the filtration procedure is considered worth the high front-end costs, for several reasons.

One reason is that it relieves the courts of the burden of making such an assessment in the first instance, a task for which they are not well equipped. Another reason is that competitors, potential investors, and the general public have certainty, or at least substantial assurance, that an issued patent covers a true invention and thus a valid, enforceable piece of intellectual property.

The certainty of a patent is further assured by patent law's requirement that a patent applicant

must define the subject matter to be protected, by use of claims describing the scope of the patent right. The claims of a patent provide a description of that which others must not make, use, or sell. By the same token, a patent's claims tell the public what is *not* staked out as the exclusive right of the patent's owner, and is therefore available to competitors.

Patent law does not protect ideas, as such, and in principle the system of protection is limited to machines and other products implementing or carrying out novel ideas. Thus, laws of nature and mathematical principles cannot be monopolized under a patent. By the same token, algorithms and computer programs, as such, cannot be patented. (A vast amount of ingenuity has been devoted, however, to obtaining patents on algorithms and computer programs; for example, by claiming them as "machine systems" or other euphemisms. This has resulted in considerable litigation.)

Other characteristics of the patent model include the right, for approximately 20 years, to exclude others from making, using, importing, and selling the patented subject matter. This patent property right against others is absolute, in the sense that independent invention is not a defense to a claim for patent infringement. (If you patent a widget, and later I independently design the same widget and start manufacturing and selling it, I am an infringer and you can shut me down.)

Finally, and most important for present purposes, a patent property right against others is additionally absolute in the sense that a patentee is free to withhold the use of the invention from others, entirely or selectively. Moreover, a patentee is ordinarily entitled not only to damages for patent infringement but also an injunction to prevent any future unconsented-to use of the invention. Subject to very narrow qualification, the US model of patent protection (unlike that of other countries) does not permit compulsory licensing or anything like it.

The copyright model

The copyright law model of legal protection requires only a very minimal level of merit or creativity—little more than failure wholly to plagiarize the work from another author. Accordingly, there is no need for examination and filtering by technical experts before legal protection attaches to a work. Ordinarily, the first time that the creative level of a work is examined is in the course of a copyright infringement suit before a federal court. The result is lower front-end cost for copyright protection, but possibly much higher costs in litigation, if that ultimately occurs.

A copyright owner has the right, typically for a 75-year term, to exclude others from reproducing, importing, and distributing the copyrighted work. Apart from control over public performance and display, however, copyright law does not prevent use of the subject matter. Thus, use of infringing software (for example, execution of an illegally copied program) is not copyright infringement unless the use involves the making of a copy. (One major court has held, however, that loading a computer program into RAM is the making of a fixed copy and thus an act of infringement. See *Micro Law*, *IEEE Micro*, June 1993.)

Independent creation of the subject matter is a complete defense, however, unlike independent invention for patents. If you write some code identical to my copyrighted code, without ever having seen my code, you are not liable to me as a copyright infringer.

Like a patent, a copyright confers an absolute property right, in the sense that a copyright owner may at will withhold from others, entirely or selectively, the right to reproduce the work. A copyright owner may secure both damages from and injunctions against those who infringe the copyright.

US law does not ordinarily permit, and the Berne Convention (a treaty to which the US recently became a signatory) prohibits, compulsory licensing

of copyrighted works. The Berne Convention also prohibits discrimination against some kinds of literary work (as which computer programs are classified) by giving them less favored treatment than other kinds of literary work.

A copyright, unlike a patent, has no claims defining its scope. Accordingly, the scope of a copyright is whatever a court ultimately holds that it is. That scope is largely unpredictable in advance. Ordinarily, a copyright will protect against verbatim copying, such as bit-for-bit or instruction-for-instruction copying of code, but that is often not the major issue in software cases. In recent years, ingenious counsel have frequently persuaded courts to protect against nonliteral, nonverbatim imitation of copyrighted works—for example, imitation of user interfaces and command languages for application programs. This has been done on the imaginative theory that computer programs and other works of new technology deserve broad protection analogous to that accorded poems and novels, for which copyright law may protect plot and other nonliteral aspects.

These developments have created great tension with the basic legal principle that copyright protects only particular expressions of ideas. Copyright does not protect any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in a work. (This list is what we may call the *index prohibitorum* of section 102(b) of the US Copyright Act. That section expressly states that the scope of copyright does not extend to the foregoing list of things.)

In allowing such "nonliteral" protection of software, some courts have confused copyrights with patents. (The *Whelan* and *Lotus-Paperback* decisions are examples of this.) Very recent decisions have tended in the opposite direction, however, and have restored the previously recognized differences between the copyright and patent

continued on p. 100

Micro Review

Richard Mateosian

2919 Forest Avenue

Berkeley, CA

94705-1310

(510) 540-7745

Programming Windows

Windows has recently grown tremendously in popularity. One large obstacle stands in the way of continued growth, and Microsoft has mounted full-scale assaults on that obstacle from several directions.

The obstacle is the steep learning curve that application programmers face when they want to work in the Windows environment. You can't write an application that displays "Hello, world" in a window without encountering arcane concepts. The new application programmer must confront the intricacies of object orientation and C++, over a thousand system calls, and the huge foundation class library.

The serious programmer's guide to this mess has been and continues to be Charles Petzold's *Programming Windows*, now in its third edition (Microsoft, 1992). This book has received universal praise, so I don't need to add anything here. If you want to program Windows applications and haven't read this book, you need to do so. Nothing in this column will change that.

Another book you should read if you want to program Windows applications is *The Windows Interface—An Application Design Guide* (Microsoft, 1992). In it, Microsoft promotes visual and functional consistency across Windows applications. Many developers hope to make their products "intuitive" to users to reduce users' resistance to learning them. These developers often forget that most of what makes a user interface intuitive is its similarity to other interfaces that the user already knows. Adhering to the standards in this book can go a long way toward making your interface easy to learn.

Visual C++, Professional Edition (Microsoft, Redmond, Wash.; \$499)

Visual C++ is a package that contains every-

thing you can get from Microsoft to program Windows applications. You can get it on CD, but the usual distribution medium is high-density diskettes—20 of them, filled with compressed files. It expands to 56 Mbytes on your hard disk. The accompanying 11 books weigh around 40 pounds.

If you're a Windows programmer with experience using C and the Windows Software Development Kit (SDK), you don't have to give them up. They're included. If you want to develop DOS applications or generate p-code or use the CodeView debugger for DOS or Windows, don't worry. It's all included.

While not abandoning the SDK, Microsoft has provided an alternate paradigm that should narrow its use. Visual C++ automates the creation of skeleton Windows applications based on the Microsoft Foundation Class Library, Version 2.0. You still need to call SDK functions as you hang flesh on the skeleton.

You invoke the Class Wizard and specify a few options in a simple dialog box, and then Visual C++ gives you a skeleton application containing a great deal of functionality. You can immediately resize or scroll your new application's window and create or save documents using your application's File menu or toolbar. You can then use the App Studio to create or modify resources like icons, menus, and toolbars. App Studio lets you manipulate these resources directly, so you rarely need to work with the resulting source files.

Your new skeleton application uses the C++ language and the Microsoft foundation classes. You can accomplish essentially the same things using C and the SDK, but C++ and the foundation classes provide a cleaner approach. For example, the constructor and destructor functions of foundation classes automatically contain calls

that you must remember to make explicitly with C and the SDK.

Visual C++ is an integrated environment with debugger, source browser, and text editor. The embedded editor uses color to emphasize language elements, and it highlights error lines after failed compilations. The downside, however, is that you can't replace the embedded editor with your own. Many programmers will suffer serious withdrawal symptoms if they can't use Brief, the outstanding text editor from Solution Systems, or Emacs, the popular Gnu editor. No doubt clever programmers will develop ways to let Brief and Emacs in the back door, and Microsoft will reluctantly incorporate those options into a future release.

Recognizing how new and how hard Visual C++ will seem to most programmers, Microsoft has bent over backwards to make it easy to use. Visual C++ has extensive on-line help. It is well organized, well written, and makes good use of automated cross-referencing (hypertext). Visual C++ also contains an excellent tutorial in which you can build a drawing program from scratch. A collection of progressively more advanced project directories allows you to check your progress in this tutorial or simply follow along without typing anything.

Visual C++ includes a 36-page "magazine," which introduces its features through interviews with members of its project team. This is an extremely helpful overview, and you should take the time to read it.

There is another way in which Visual C++ is easy to use. Microsoft C development systems have been notoriously hard to install and get running. When I reviewed the Borland C++ package (August 1992), I wasn't able to include a review of the Microsoft C development system, because I've never been able to get it to compile even the sample programs that came with it. I didn't have that problem with Visual C++. It didn't work immediately, but I was able to get it running with small changes in my DOS config.sys file.

This is a monumental software package, but if you intend to create Windows applications, this is the software you should use to do it.

Inside Visual C++, David J. Kruglinski (Microsoft Press, Redmond, Wash., 1993, 631 pp. plus diskette; \$39.95)

Early press releases referred to this book by the cumbersome but helpful title *The Microsoft Guide to C++ Programming in Windows*. The current title misrepresents the book a little, since it doesn't contain much information about the inner workings of Visual C++, while it does say a lot about C++ programming for Windows with the foundation class library. It doesn't replace Petzold's book, but a synthesis of the two will surely be forthcoming.

Kruglinski recognizes that programmers have purchased far more C++ books than they have read. He includes a 32-page summary of what he considers the essentials of C++ that you need to understand to read this book. I thought it was good, but his editors made him call it "a personal view."

I think the most useful thing about the book, justifying its title, is the way it leads you through all of the ins and outs of using Visual C++. If you want to go beyond the Visual C++ tutorial, or if you just find 40 pounds of manuals daunting, get this book and use it as your guide.

Visual Basic 3.0, Professional Edition (Microsoft, Redmond, Wash.; \$495)

Basic began as a limited teaching language in the 1960s. In the mid 1970s and early 1980s, Microsoft's version of Basic became a nearly universal language for personal computers. It led to a proliferation of amateur programs. These programs arose quickly and easily from user needs. Small Basic programs are easy to write and understand. However, they tend to grow until they become unmaintainable. Basic is now largely obsolete as a programming language, but Microsoft uses it for special

purposes, such as the macro language for Word for Windows.

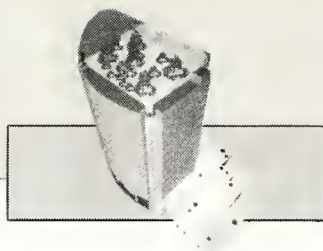
The event-driven structure that underlies Windows leads to the need for a large number of small routines. Visual Basic takes advantage of this fact. Users construct the application interface through direct manipulation of graphic elements, like buttons and menus, then associate Basic programs with them. Each event that an object needs to handle gives rise to a Basic subroutine. The detailed functionality is in the small programs, while the system retains control of the interface elements that the user specified.

As described, Visual Basic sounds like a prototyping tool or a vehicle for relatively limited programs. However, Microsoft has made it possible for you to build powerful systems with Visual Basic. You organize Visual Basic applications into projects. The graphic elements with associated Basic programs are called forms. In addition to forms, you can include code modules. These allow you to build subroutines that you can call from more than one form.

Of course, the code modules are an open invitation to the kind of amateur programming described earlier. Recognizing this, Microsoft has provided a number of improvements to Basic to help you avoid the kind of spaghetti code and global scopes of original Basic. The most important drawbacks of original Basic—line numbers and the GOSUB command—are long gone. Instead, you can use the same kinds of control structures that you have in other languages. You can even declare a procedure to be private to its code module. If you're a real purist, you can tell Basic to require you to declare all variables before you use them.

Another way to extend the power of Visual Basic is to link your program to procedures in dynamically linked libraries (DLLs). You can write your own DLL procedures—presumably in C, C++, or assembly language—or you can use any Windows system DLL pro-

continued on p. 99



Guest Editor's Introduction: **Toward a World Filled with Computers**

Ken Sakamura

University of Tokyo

It is my pleasure to present another East Asia issue, the first in two years. In my Guest Editor's Introduction of 1991¹ I noted some differences between the North American and Japanese microelectronics research and development scenes. The United States tended to emphasize high-powered applications such as superfast workstations and gigabit networks. In contrast, Japan gave more attention to the need for cost performance in home electronics and other consumer-oriented products.

That was two years ago. Today, however, that clear-cut distinction is changing. True, we still see continued competition over development of high-performance CPUs in the 100-MIPS-plus class for workstations and personal computers, with little regard for power consumption or chip size. At the same time, however, growing importance is being given around the world to microprocessors in the 10-MIPS range. These processors—which along with high performance feature low-power dissipation and small chip size—support embedded systems. We are seeing the appearance of high MIPS-per-watt microprocessors. Ever since the collapse of the Berlin Wall between East and West, emphasis has been shifting away from military technology toward civilian electronics markets. Even the United States is pouring new energy into embedded processors for portable systems and other consumer products. The applications for high-performance microprocessors are beginning to change dramatically.

In the TRON Project that we have been promoting over the last several years, we have continued to describe the future computer society as evolving along certain lines. We see the objects that surround us in our daily lives becoming increasingly embedded with computer chips, sen-

sors, and actuators. These computerized objects then become linked by wired and wireless networks, forming distributed-processing systems in which the objects collaborate with each other.² (See Figure 1.)

We are now well into the 1990s, and things are moving increasingly in the direction of our predicted scenario. Recently, terms like *computer-augmented environment* and *ubiquitous computing*³ seem to have become buzzwords in computer science. They describe an environment in which computers are used everywhere around us. This trend reflects the advances in development of the constituent technologies, such as small but high-performance sensors and displays, and high-performance sensors with low-power dissipation. It is becoming quite feasible to realize a world filled with computers.

In the next few years the world of microelectronics is likely to undergo great changes. The time has come to look ahead to applications for a world full of computers. With this background in mind, I describe recent microprocessor developments in Japan and, as in past issues, bring readers up to date on the TRON Project.

Japan's microprocessors

It is still true that most originally developed processors in Japan are designed for embedded system use. Japan has a number of general semiconductor manufacturers, including NEC, Toshiba, Hitachi, Fujitsu, and Mitsubishi Electric. But when it comes to microprocessors for workstations and personal computers, the manufacturers are limited to licensed production of the architectures of US companies (especially RISC chips). In view of the demand for RISC chips, none of the Japanese firms is developing its own original chip for

workstation or personal computer use. Moreover, they are hesitant to attempt developing chips compatible with the Intel x86 family because of the potential for legal trouble. The result is a further strengthening of the trend toward chips for embedded systems.

A boom in CPU development for low-power equipment persuaded Japanese manufacturers to develop original-architecture microprocessors (MPUs) and microcontrollers (MCUs). But up to now these were nearly all 4-, 8-, or 16-bit products. (In terms of quantity the 4-bit chips are most common.) Practically the only 32-bit chips to date have been the TRON-specification chips developed by six firms and the NEC V series.

Since last year however, 32-bit MPUs and MCUs aimed at large-scale use in portable systems have increased. ARM and Hobbitt are well known outside Japan. Since 1992 various firms in Japan have announced 32-bit chips with low-power consumption aimed at the portable equipment market. The chips typically perform at 10 to 20 MIPS (in terms of VAX MIPS; that is, VAX 11/780 performance on Dhrystone 1.1 or 2.1 benchmarks). Their power dissipates at under 500 mW; the CPU core size is less than 50 mm². Specific applications include highly portable personal information systems and game machines.

The instruction set architecture of these chips typically blends RISC and CISC approaches, and a 16-bit instruction format improves object code efficiency. In other words, rather than going all out for performance as a pure RISC, Japan's manufacturers balance performance needs with the needs for small object code size and higher code density. Architecturally, the design is conservative, but applications of state-of-the-art technology realize the advantages of both RISC and CISC approaches.

It is this reworking of processor design that deserves our attention. Because of the emphasis on preserving past computer system investment, the companies have long been reluctant to depart from architectures like the IBM 370 or Intel x86. But in the case of processors for embedded systems, they have found it relatively easy to switch over to a new microprocessor design each time new equipment is introduced. When they compare the new designs to larger computer systems, they are less concerned about past investment. Makers of VCRs or copiers, for example, who are embedded-MPU users, care little about whether past software investment or yesterday's object code will run on a new system. Their concern is for deriving the maximum cost performance available at a given point in time. Thus they can readily adopt a processor with a new architecture.

In the latter half of 1992, NEC's V800 series and Hitachi's SH7000 series were announced as low-power 32-bit microprocessors. Then in the first half of 1993, Toshiba's TX2 and Mitsubishi's M16 were announced as second-generation TRON-specification, low-power-consumption processors for embedded systems. Common to each of these products is

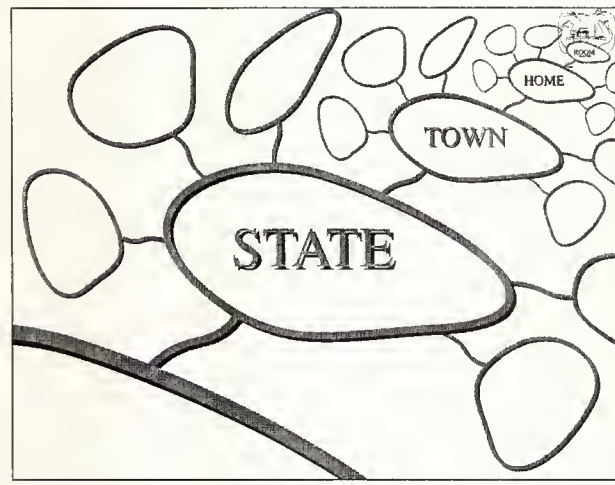
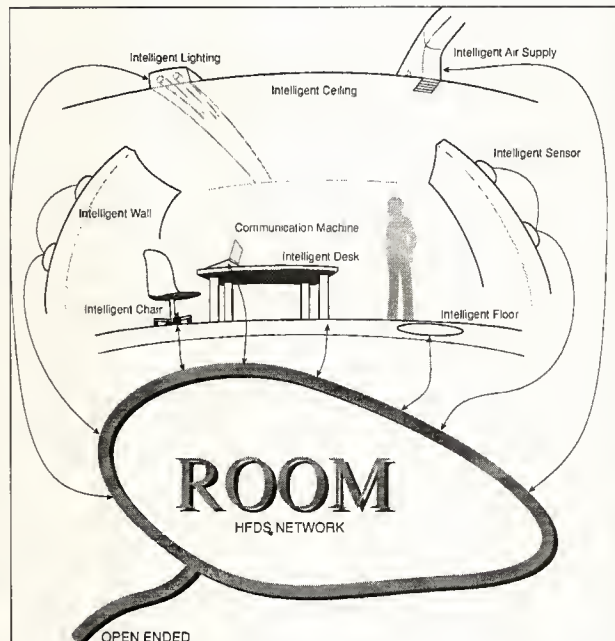


Figure 1. Intelligent objects and their networks.

that their manufacturers are making available ITRON-specification real-time operating systems. Table 1, next page, outlines the features of each new processor.

High-performance microprocessor development continues as well. Japanese manufacturers are presently gearing up to market three high-performance processors with original designs. Currently under development are Hitachi's Gmicro/500, Fujitsu's μ VP, and Mitsubishi's Gmicro/400. The last of these is a high-speed, 32-bit processor for embedded systems that supports only integer calculations.

Table 1. The 32-bit, low-power-consumption processors in Japan.

Processor	Features
NEC V810	<p>32-bit, low-power-consumption microprocessor</p> <p>2.2V to 5V power supply; 15-MIPS performance at 5V and 25 MHz with 500-mW power dissipation and 40-mW power consumption at 2.2V and 10 MHz</p> <p>32 general registers of 32 bits each; support for both 16-bit and 32-bit instruction formats, for efficient object code density</p> <p>Bit-string manipulation instructions; single-precision floating-point operation instructions conforming to IEEE Std. 754</p> <p>0.8-μm CMOS technology; 240,000 transistors on a 7.7\times7.7-mm chip; an on-chip 1-Kbyte instruction cache</p> <p>RX732 ITRON-specification real-time operating system</p> <p>16-bit external bus version (V805) available</p>
Hitachi SH7000 series	<p>32-bit microcontroller family with on-chip DSP facility</p> <p>16-MIPS performance at 20 MHz with a 5V power supply</p> <p>On-chip, 16-bit hardware multiplier unit for (16 bits \times 16 bits + 42 bits) 42-bit multiplication and accumulation operation performed in 100 ns to 150 ns</p> <p>500-mW power consumption, or 100 mW with a 3V power supply</p> <p>0.8-μm CMOS process with a CPU core integrating 38,500 transistors in an area of 6.58 mm²</p> <p>Model SH7032 with peripheral functions and on-chip memory integrates approximately 593,000 elements on a 10.78 \times 10.1-mm chip</p> <p>16 general registers; 16-bit fixed-instruction format</p> <p>μITRON-specification operating system available</p>
Toshiba TX2	<p>32-bit, TRON-specification, low-power-consumption microprocessor</p> <p>Approximately 14-MIPS performance at 25 MHz (5V power supply), 1.7 times faster than the first-generation TX1</p> <p>1/10th power consumption cut during WAIT instruction execution</p> <p>16-bit instruction execution in one clock cycle</p> <p>Chip size one half that of the TX1</p>
Mitsubishi M16	<p>TRON-specification microcontroller with 16-bit external bus and 32-bit internal bus</p> <p>4- to 5-MIPS average performance at 10-MHz operation (5V power supply)</p> <p>16-bit instruction execution in one clock cycle</p> <p>Typical configuration (M31000S2FP) integrates 2 Kbytes of memory and peripheral functions</p>

These are the only general-purpose microprocessors with an original architecture. As mentioned earlier, other products are being developed that license the RISC architectures of US firms (Sparc, HP-PA, Alpha, MIPS).

Hitachi Gmicro/500. This 32-bit, TRON-specification microprocessor adopts a superscalar architecture that permits 130-MIPS performance at 66 MHz. When compared to Intel's 66-MHz Pentium, this chip dissipates only a third as much power, is a third smaller, and is 20 percent faster. Its designers used a 0.6- μ m CMOS process.

How is it that the Gmicro/500 boasts performance surpassing Intel's Pentium? The answer lies in its being targeted mainly for embedded system use. Because of Intel's lock on the personal computer market, the Gmicro/500 had no other choice but to aim for use in embedded control systems. And the only way to win in the high-performance embedded system market was to focus on small chip size, low-power consumption, and high speed. In other words, its excellence is the result of its being aimed not at personal computer and workstation use but at embedded systems.

This microprocessor also, of course, runs an ITRON-specification operating system. In addition, designers are developing a BTRON2-specification operating system for workstation use. The latter operating system supports a hypertext structure at the operating system level, as well as offering functions for multimedia support, multilingual processing, and distributed processing; it provides a compact kernel.

Fujitsu μ VP. The μ VP is a vector-processing-architecture coprocessor for direct connection to TRON-specification 32-bit microprocessors. At 50 MHz, 206-Mflops single-precision speed, and 106-Mflops double-precision speed, this 0.5- μ m CMOS chip performs on a par with early supercomputers.

Separate articles on these two products appear elsewhere in this special issue. They are examples of how manufacturers continue to take up the challenge of developing a Japanese-original chip running an originally developed operating system, and are steadily making progress in implementing the goal.

There was a time when people thought the future belonged only to RISC and that CISC chips would disappear. But in Japan the emphasis is on object efficiency. For this reason a blend of RISC and CISC designs was sought. As a result, people have come to appreciate an architecture like that of the TRON-specification chips; new microprocessors implementing this architecture are being developed today. Some of the applications for which they are being used are high-performance numerical control machines and communication control processors. Then we have the μ VP coprocessor with its vector-processing approach, a unique product that stands out among the peripheral chips designed for the TRON architecture. A single-board computer running the Gmicro/500 and μ VP is said to achieve performance equivalent to that of a Cray 1 supercomputer, demonstrating amazing progress in microelectronics technology.

TRON Project update

The TRON Project is looking ahead to a world filled with computers. As I noted at the beginning of this piece, interest is beginning to focus on environments that are filled with computers everywhere you look. The TRON Project has constructed a pilot TRON-concept Intelligent House^{4,5} with around 1,000 built-in computer elements. We are about to begin construction of the first TRON-concept Intelligent Building that incorporates tens of thousands of computers.

What sets the TRON Project apart is its attempt to get a jump on the next computer age by considering what kinds of computer applications are likely to emerge, actually building such applications, and feeding back the results into the design of basic components such as microprocessors and operating systems.

Besides trying to determine the most suitable architecture for an age when the number of microchips in use is thousands or ten thousands of times greater than today, this project

takes a comprehensive look at questions such as the following. What should computers be made to do? What should they not be made to do? What kinds of infrastructures are needed? What rules are necessary for data interchange? In these ways the project is planning and building information infrastructures for the future.⁶

From the TRON Intelligent House to the TRON Hyper-Intelligent Building. We completed the TRON-concept Intelligent House experiment, conducted as an application project to get an advance look at the future, in the spring of 1993. At the same time we finished the basic design of the TRON Hyper-Intelligent Building, incorporating tens of thousands of computer elements, in preparation for the start of construction later this year. (See Figure 2, next page.)

The significance of this project is that a life-size model of a "computers everywhere" building will be built and put to actual use as a place of work. The building's computers will be able to locate people wherever they go and will fine-tune lighting (see Figure 3), temperature, and other environmental factors to personal preferences. The overall model is that described earlier of "intelligent objects" (ordinary objects containing microchips, sensors, and actuators) linked in wired or wireless networks that enable them to coordinate their actions. The component parts making up these networks are the results of fundamental research and development taking place in the TRON Project. In addition to the microprocessors⁷ used to control the intelligent objects, these important development results include real-time operating systems, high-level data interchange protocol, and human-machine interface specifications.

The importance of open architecture. One of the vital requirements of a processor in intelligent objects is outstanding real-time performance. Important likewise are the ability to be used as an ASIC core, the possibility of applying the same architecture to a whole range of processors from low-power-consumption models to high-end products, and the adoption of an open-architecture policy so that anyone is free to develop compatible microprocessors. When intelligent objects become networked in the tens of thousands or millions, it would be highly unlikely that all the component parts could be supplied by one corporation. In such an age, the basic architectures, operating system interfaces, data interchange protocols, and other basic technologies will have to be open to all as social infrastructure. Success is not possible under the dominance of any one company. I believe this open-system approach needs thoughtful consideration.

Subprojects. I have already touched on the status of the TRON-specification microprocessors. Other fundamental subprojects are currently under way.

ITRON. The ITRON-specification real-time operating systems for embedded systems have been implemented for most Japanese microcontrollers, and it has become a de facto industry standard. The ITRON standards continually undergo

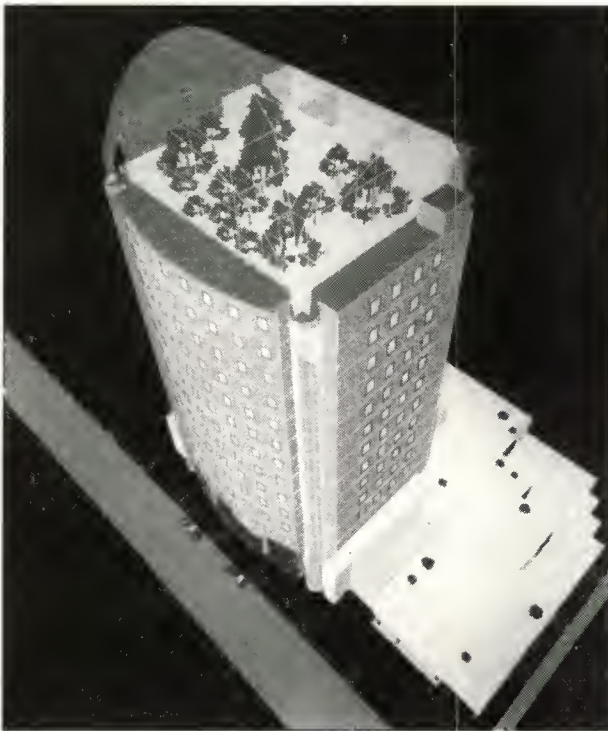


Figure 2. The TRON Hyper-Intelligent Building. © Ken Sakamura 1990.

improvements; the newest version called μ ITRON 3.0 offers network support. That is, we extended the specification to permit application to distributed systems connected in loosely coupled networks. When CPU nodes running μ ITRON 3.0 are networked, programmers can use ordinary system calls to manipulate tasks or semaphores in other nodes. Naturally, the specifications are open. Any interested readers can obtain copies of the English-language specifications via Internet's

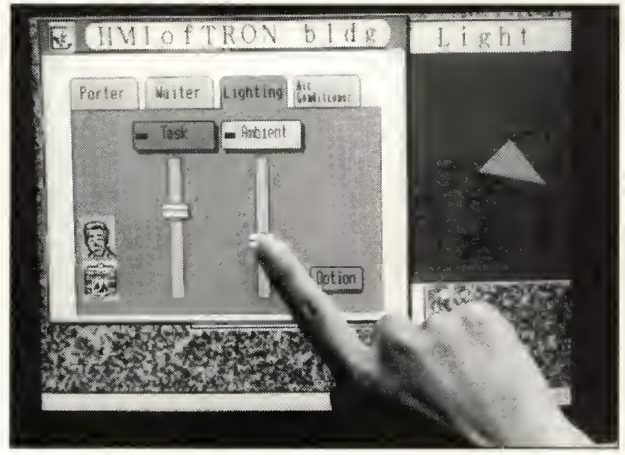


Figure 3. A BTRON screen panel for lighting control. © Ken Sakamura 1990.

anonymous ftp utsun.s.u-tokyo.ac.jp; the directory is /TRON/ITRON/SPEC.


BTRON. When the ubiquitous computer age finally arrives, failure to standardize a human-machine interface (HMI) will invite confusion. The BTRON subproject is not limited to personal computers but deals with this interface in a much broader sense. The BTRON operating system specifications define a compact, highly efficient graphical user interface-based operating system designed for use even in light switches, of the kind adopted in the TRON-concept Intelligent House (see Figure 1). Compared to Microsoft Windows, it runs at high speed with minimal resources. (A workable system has been implemented around an i286 processor running at 16 MHz, with 4 Mbytes of memory and a floppy-disk drive.)

BTRON's fully multitasking operating system also supports a hypertext/hypermedia function at the system level. Since it would hardly be feasible to apply the GUI approach to all the switches around us, the TRON HMI specifications⁸ also describe non-GUIs, such as physical switches and handles (called SUI). Also specified in the BTRON subproject is the TAD (TRON application databus) high-level data interchange protocol for use among intelligent objects. This protocol is starting to be used in systems consisting of intelligent objects linked by a simple token-ring bus (a μ BTRON-specification bus⁹), for system operation and the like, to achieve a consistent operation environment.

CTRON. These specifications apply to operating systems used in communications systems and servers such as big telephone companies and private branch exchanges. In 1993 Tandem Computers announced a nonstop computer system running on a CTRON-based operating system.

TRON progress. As this summary should indicate, the TRON Project is proceeding at a steady pace, making its mark

as an original Japanese computer development project. The standards and specifications produced by the TRON Project are openly available to anyone by writing to TRON Association, Katsuta Building 5F, 3-39, Mita 1-chome, Minato-ku, Tokyo 108, Japan. (The contact address for North America is TRON Association US Liaison Office, PO Box 23990, Tempe, AZ 85285.) Moreover, the standards can be used freely, without licensing fees.

SINCE READERS ARE ESPECIALLY INTERESTED in microprocessors, I have prepared articles in this special East Asia issue on two of the chips I've mentioned: the general-purpose microprocessor Gmicro/500 and the "single-chip supercomputer" μ VP coprocessor. Many other unusual microprocessors are being developed in Japan, but of these I have chosen to present a high-speed fuzzy chip. I hope you enjoy reading these three articles. 

References

1. Ken Sakamura, "Guest Editor's Introduction," *IEEE Micro*, Aug. 1991, pp. 12-15.
2. Ken Sakamura, "The Objectives of the TRON Project," *Proc. Third TRON Project Symp.*, Springer-Verlag, Tokyo, 1987, pp. 3-16.
3. Special Issue: Computer Augmented Environments, *Comm. ACM*, July 1993.
4. Ken Sakamura, "About the Cover: The TRON Intelligent House," *IEEE Micro*, Apr. 1990, pp. 6-7.
5. Volker Hartkopt et al., *Designing the Office of the Future: The Japanese Approach to Tomorrow's Workplace*, John Wiley & Sons, New York, 1993.
6. Ken Sakamura, "TRON Application Projects: Gearing Up for HFDS," *Proc. Eighth TRON Project Symp.*, IEEE Computer Society Press, Los Alamitos, Calif., 1991, pp. 2-14.
7. Ken Sakamura, *An Introduction to T-Open*, TRON Association, Tokyo, 1992.
8. TRON Electronic Equipment HMI Research Group, *TRON Human-Machine Interface Specifications*, TRON Association, 1993.
9. Ken Sakamura et al., "The μ BTRON Bus: Functions and Applications," *Proc. Third TRON Project Symp.*, Springer-Verlag, 1989, pp. 101-112.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 150

Medium 151

High 153

COMING IN DECEMBER

IEEE Micro Special Issue on Standards

December's feature articles explore the organizational and economic aspects of designing and constructing standards, those practical tools that help us make tools. Guest Editor Stephen L. Diamond comments, "Without formal or informal standards, we can create nothing except that which operates in isolation. Standards provide an interface, a means of communication that lets two disconnected and disparate things work together as one." When they are developed correctly, standards provide this bridge.

Don't miss reading:

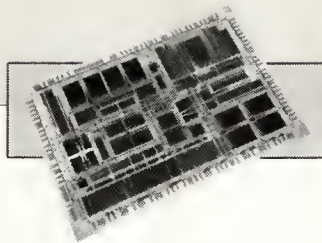
- **The Art of Building a Market with Standards**
- **The War of Words: Intellectual Property Laws and Standardization**
- **Standard Setting in the US: Public and Private Sector Roles**
- **Forming, Funding, and Operating Standard-Setting Consortia**
- **Standardization and the Information Infrastructure**
- **and more**

An extra treat:

December's Micro View features an interview with John Crawford of Intel Corporation in which he discusses the Pentium and the market. Don't miss this one!

Be sure to read the December 1993 issue of

IEEE **MICRO**



The Gmicro/500 Superscalar Microprocessor with Branch Buffers

Featuring a RISC-like dual-pipeline structure for high-speed execution of basic instructions, the Gmicro/500 represents a significant advance for the TRON architecture. Upwardly object-compatible with earlier members of the Gmicro series, this microprocessor uses resident dedicated branch buffers to greatly enhance branch instruction execution speed. Also, its microprograms simultaneously employ dual execution blocks to effectively execute high-level language instructions. Fabricated with a 0.6- μ m CMOS technology on a 10.9 \times 16-mm die, the chip operates at 50/66 MHz and achieves a processing rate of 100/132 MIPS.

Kunio Uchiyama

Fumio Arakawa

Susumu Narita

Hirokazu Aoki

Ikuya Kawasaki

Shigezumi Matsui

Mitsuyoshi Yamamoto

Norio Nakagawa

Ikuo Kudo

Hitachi, Ltd.

As one of the new high-end microprocessors that perform at over 100 million instructions per second,¹⁻⁴ the Gmicro/500 builds upon the well-established TRON architecture.⁵ The TRON specification includes 135 variable-length instructions with 14 addressing modes. Operating at 50/66 MHz, this chip is upwardly object-code compatible with earlier members of the Gmicro series.⁶⁻⁹ To achieve a processing rate of 100/132 MIPS, it uses a sophisticated superscalar technique and high-speed branch accelerators optimized to a complex instruction-set computer architecture.^{10, 11}

We established the performance target for the Gmicro/500 at over 100 MIPS with two constraints in mind: object-code compatibility and cost. The Gmicro microprocessors already have many customers in such diverse fields as industrial equipment, communication switching systems, fault-tolerant computers, and business workstations. To retain this existing customer base, the Gmicro/500 must support the TRON architecture.

This well-established architecture features variable-length instructions and includes rich, high-level instructions, such as string, variable-length bit field, multiple-word load/store, and operating system-related instructions. We developed a new superscalar method for this CISC architec-

ture that can execute two instructions simultaneously. Our technique can also effectively process high-level instructions using two pipelines.

In microprocessor design, cost depends on clock frequency and power dissipation. A high clock frequency increases power dissipation, which in turn necessitates costly large-scale integration packaging and expensive system cooling. A high clock frequency also requires high-speed external memory, further increasing system costs.

To improve performance without relying solely on an increased clock frequency, we sought to reduce the effective CPI, or clocks per instruction, to 0.5. In other words, we wanted our system to execute two instructions per clock. To achieve this goal, we needed to develop not only a sophisticated superscalar technique but also high-speed branch accelerators that reduce branch penalties.

In introducing the Gmicro/500, we first describe the chip's specifications and internal architecture. Included with this is a discussion of the new sophisticated superscalar technique and high-speed branch accelerators that are optimized to a CISC architecture. Afterwards we will present the results of a performance analysis on this new microprocessor.

Table 1. Gmicro/500 specifications.

Parameter	Specification
Clock frequency	50/66 MHz
Performance	100 MIPS (50 MHz) 132 MIPS (66 MHz)
Basic CPI	0.5 clocks (two instructions per second)
Voltage	5 V
Power dissipation	7 W (50 MHz), 9 W (66 MHz)
Package	Ceramic QFP, 256 pins
Technology	0.6- μ m CMOS, three metal layers
Die size	10.9 \times 16.0 mm
Instruction cache	8 Kbyte, 2 way
Operand cache	8 Kbyte, 2 way, write through
Instruction TLB	64 entries, 2 way
Operand TLB	64 entries, 2 way
Branch target buffer	64 entries, 2 way
Return buffer	8 entries
Store buffer	4 entries by 8 bytes
External address bus	32 bits
External data bus	64 bits

Chip specifications

Table 1 summarizes the Gmicro/500 specifications. We used a three-metal-layer, 0.6- μ m complementary metal-oxide semiconductor technology for its fabrication. Shown in Figure 1, this microprocessor integrates 1.65 million transistors on a 10.9 \times 16-mm die. The chip incorporates a RISC-like super-scalar integer execution unit, resident two-way set-associative 8-Kbyte instruction and operand caches, and a 64-entry branch target buffer.

The Gmicro/500 also has an eight-entry return buffer, a four-entry store buffer, and a memory management unit with comprehensive section and page protection. Its floating-point processing unit—fully compatible with the ANSI/IEEE 754-1985 standard—operates in parallel with the integer execution unit. As shown in Figure 2, the Gmicro/500 achieved 100/132 MIPS at 50/66 MHz—the top performance of recent CISC microprocessors.

Internal architecture

As shown in the block diagram of Figure 3 (next page), the Gmicro/500 consists of five basic units:

- Instruction
- Decode
- Execution
- Access
- Floating-point processing

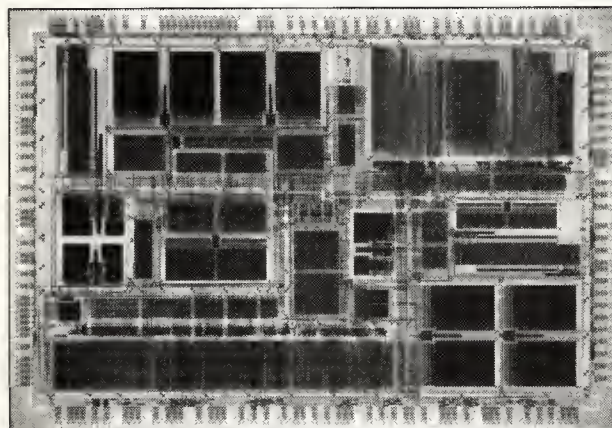


Figure 1. Photomicrograph of the Gmicro/500.

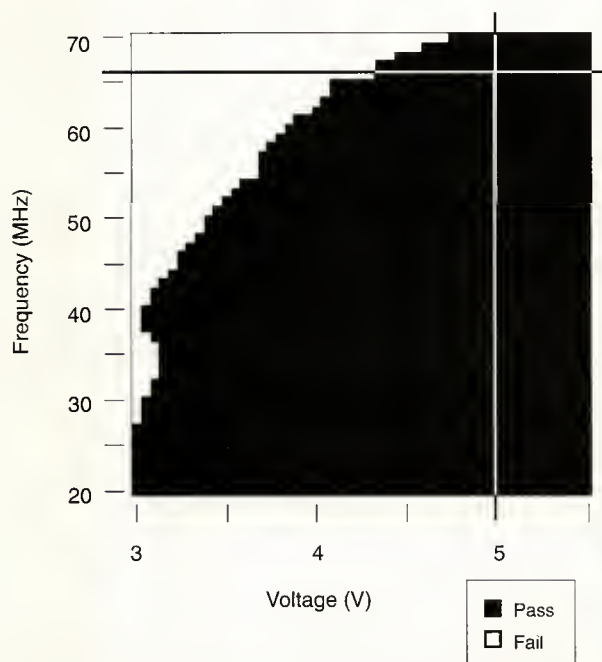


Figure 2. Schmoo plot.

Using the prefetch address generator, the instruction unit prefetches instructions from the instruction cache or external memory. These it transfers to the instruction prefetch queue. The prefetch queue consists of three sets of 64-bit latches. In addition to the 64-entry instruction translation look-aside buffer and 8-Kbyte instruction cache, the instruction unit contains a 64-entry branch target buffer and an eight-entry return buffer. These are dedicated to caching branch information for accelerated branch execution.

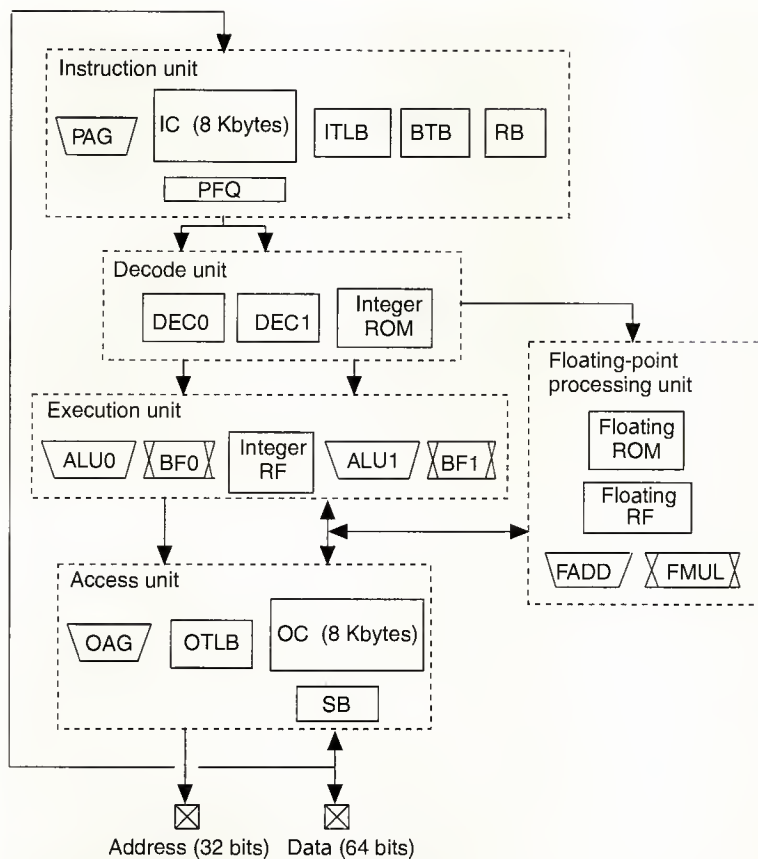


Figure 3. Gmicro/500 block diagram.

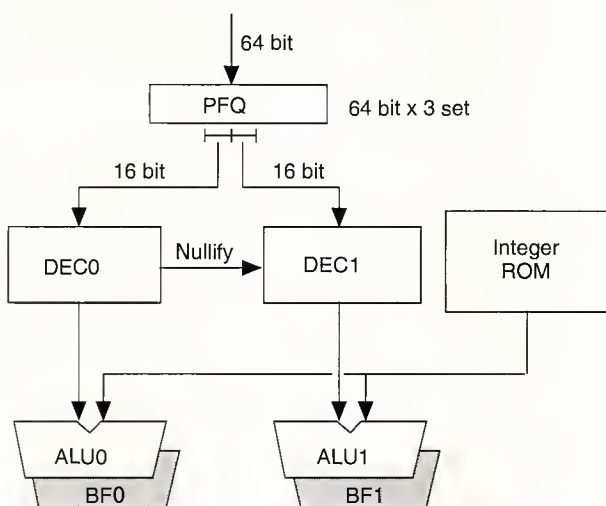


Figure 5. Decoding scheme.

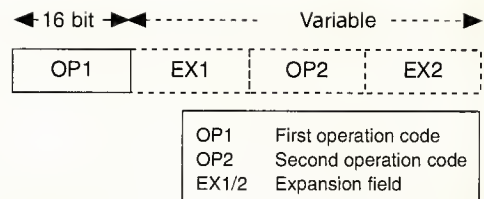


Figure 4. Instruction format for TRON specifications.

The decode unit decodes instructions using two 16-bit decoders (DEC0, DEC1). Because the length of instruction code can vary (see Figure 4), the simultaneous decoding of two instructions presents a major difficulty. The Gmicro/500 therefore decodes two instructions simultaneously only when the instruction decoded by DEC0 is 16 bits wide. As shown in Figure 5, DEC0 and DEC1 decode two successive 16-bit lengths that are extracted from the prefetch queue. If the information decoded by DEC0 is not a 16-bit-wide instruction, the result decoded by DEC1 is nullified.

In the Gmicro/500 instruction set, all basic instructions may be coded in a special 16-bit short format; compilers frequently use these short instructions. Table 2 lists the appearance probability of the field that follows the 16-bit first operation code (OP1) in five benchmark programs. The high probability of the OP1 field that follows the previous OP1 means that the simultaneous decoding by the two decoders has a high probability of succeeding. With high-percentage use of the 16-bit instruction format, the dual-pipeline decoding scheme significantly reduces implementation complexity without sacrificing performance.

Table 2. Appearance probability of the field following OP1.

Program	Type		
	OP1	OP2	EX
Dhrystone	0.63	0.06	0.31
EDN-E	0.86	0.00	0.14
EDN-F	0.65	0.12	0.23
EDN-H	0.81	0.00	0.19
EDN-K	0.67	0.16	0.17
Average	0.72	0.07	0.21

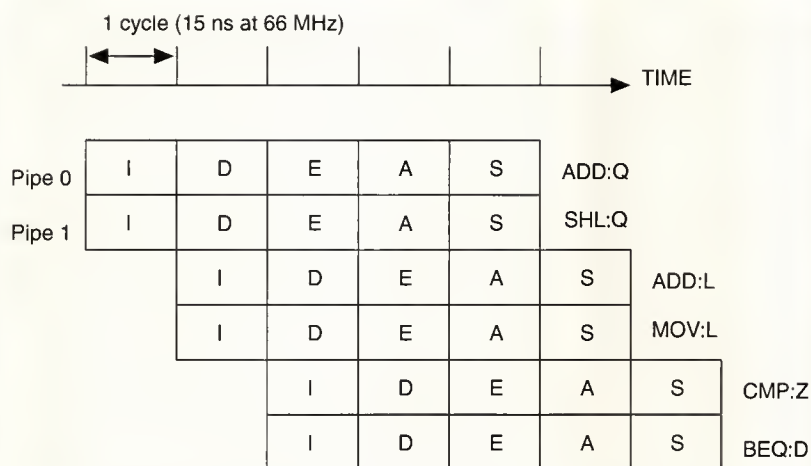


Figure 6. Dual pipelines.

The decode unit also includes an integer ROM that stores the microprograms for high-level instructions, such as string, variable-length bit field, and operating system-related instructions. One of the two decoders controls the first execution cycle of the high-level instruction, and the 120-bit microinstructions from the integer ROM control the next execution cycles. Simultaneous use of dual arithmetic logic units and dual barrel shifters in the execution unit in the microprograms enables the Gmicro/500 to execute high-level instructions quickly.

The execution unit contains the ALUs (ALU0, ALU1), the barrel shifters (BF0, BF1), and an integer register file that has four read and two write ports. Using these duplicated resources, the execution unit can execute two basic integer instructions at the same time. It can execute high-level instructions by using the dual ALUs to concatenate and manipulate data in 64-bit units. (A high-level instruction is one that executes operations that would normally require several basic machine-code instructions.)

The floating-point processing unit contains a 55-bit floating-point adder, a 66-bit floating-point multiplier, a floating-point ROM, and a floating register file with one write and two read ports. Because the floating ROM independently controls the floating-point processing unit, the processor can execute floating-point instructions in parallel with integer instructions.

The remaining unit of the Gmicro/500 is the access unit, used to control operand access and external bus cycles. It includes a 64-entry operand TLB, an 8-Kbyte operand cache that supports a write-through protocol, a four-entry store buffer, and an operand address generator. The access unit connects to the outside through a 32-bit address bus and a 64-bit data bus.

The block size of the instruction cache and operand cache

Commonly used acronyms

ALU	Arithmetic logic unit
AU	Access unit
BRA	Branch always
BSR	Branch to subroutine
BTB	Branch target buffer
FADD	Floating-point adder
FILO	First in, last out
FMUL	Floating-point multiplier
IC	Instruction cache
ITLB	Instruction TLB
OAG	Operand address generator
OC	Operand cache
OTLB	Operand TLB
PAG	Prefetch address generator
PFQ	Prefetch queue
RB	Return buffer
RTS	Return from subroutine
SB	Store buffer
TLB	Translation look-aside buffer

is 32 bytes. The fast cache fill can be done by burst transfer at 6.4 bytes per cycle. To keep cache coherency in multiprocessor systems, the Gmicro/500 automatically snoops the external bus address and invalidates the hit blocks in the instruction cache and operand cache. Since dual-port RAMs are used at the tag sections of both the instruction and operand caches, the bus snoop access to the caches can occur simultaneously with operation of the normal cache access. Using the operand address generator in the access unit, the hardware in the chip automatically handles the TLB miss, enabling the Gmicro/500 to minimize the TLB miss overhead.

Superscalar pipelining

The architecture of the Gmicro/500 builds upon sophisticated superscalar techniques.

Dual pipeline. The Gmicro/500 has dual-integer pipelines. As Figure 6 shows, each of these is divided into five stages:

- Instruction prefetch (I)
- Instruction decode (D)
- Operation execution (E)
- Memory access (A)
- Register store (S)

This structure allows the simultaneous execution of two basic integer instructions. The RISC-like superscalar pipeline structure optimizes the execution speed of register-register and basic load/store operations. Other operations (such as

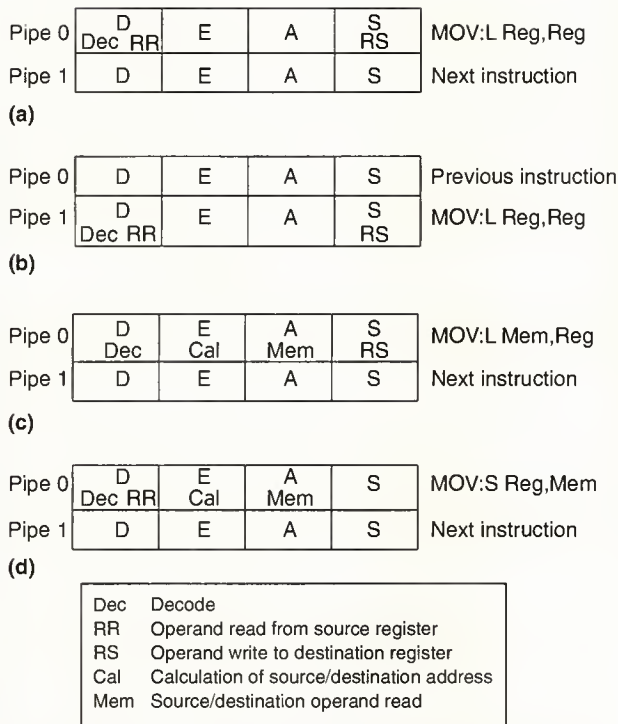


Figure 7. Pipeline stage of transfer instructions: MOV:L Reg,Reg on pipe 0 (a); MOV:L Reg,Reg on pipe 1 (b); MOV:L Mem,Reg on pipe 0 (c); MOV:S Reg,Mem on pipe 0 (d).

string and multiple-word load/store instructions) use the dual ALUs to increase execution speed by concatenating and operating on data in 64-bit units.

Figure 6 also shows a basic example of integer instruction execution flow. The pipeline streams pipe 0 and pipe 1 fetch ADD:Q and SHL:Q, respectively. The processor's decode unit decodes these two instructions simultaneously, then its instruction unit fetches the following ADD:L and MOV:L. When ADD:Q and SHL:Q are in the operation execution stage, the processor simultaneously decodes ADD:L and MOV:L, then fetches CMP:Z and BEQ:D. In this manner, the pipeline stages of 10 instructions run concurrently in five pipe-pair sequences every clock cycle, effectively completing two instructions per clock cycle.

Cache misses and register conflicts may cause insertion of wait cycles into the pipeline flow. Other obstacles (for example, branch operations) also pose potential hazards to the pipeline flow. Furthermore, high-level language instructions executed directly under microprogram control require a sequence of stages that extends over more than one pipe. The Gmicro/500 incorporates special features to minimize, and in some cases eliminate, obstacles to the pipeline flow, thus ensuring high performance for all types of programs.

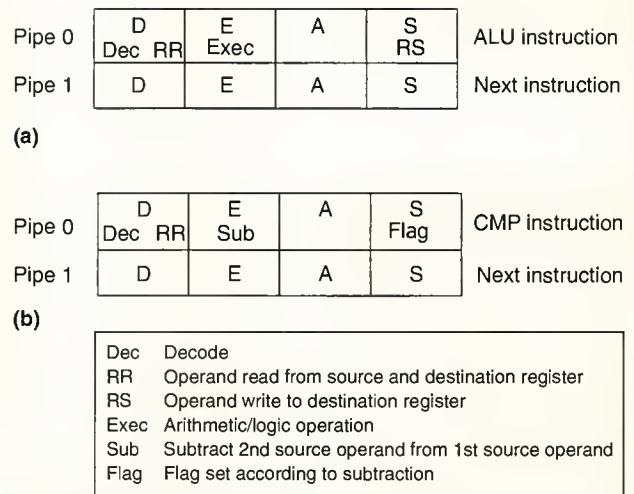


Figure 8. Execution of instructions: arithmetic/logic instructions (a); compare instructions (b).

Pipeline stage for transfer instructions. Figure 7 outlines the execution flow for transfer instructions. Register-register operations appear in Figure 7a and b. In stage D, the processor decodes the instruction and reads the source operand from the designated register. It then loads the source operand into the destination register in stage S, completing the transfer operation. Stages E and A are vacant—unused—in this type of operation. Figure 7a shows pipeline flow when the processor is executing MOV:L Reg,Reg operation on pipe 0, allowing simultaneous execution of the next instruction on pipe 1. In Figure 7b, the MOV:L Reg,Reg instruction is executed on pipe 1 at the same time that the previous instruction is executed on pipe 0.

Figure 7c shows the execution of a load operation, with the MOV:L Mem,Reg instruction executed on pipe 0 and the next instruction simultaneously executed on pipe 1. In this example, the processor decodes MOV:L Mem,Reg instruction in stage D of pipe 0 and calculates the effective address of the source operand in stage E. It then reads the source operand in stage A and loads it into the destination register in stage S.

Figure 7d shows a store operation in which the processor executes the MOV:S Reg,Mem instruction on pipe 0 while simultaneously executing the next instruction on pipe 1. In stage D on pipe 0, the processor decodes the MOV:S Reg,Mem instruction and reads the source operand from the designated register. It calculates the effective address of the destination location in stage E and stores the source operand in the addressed memory location in stage A, completing the transfer operation. In this example, stage S is vacant (unused).

Execution of arithmetic/logic instructions. Figure 8a shows the execution flow of arithmetic/logic instructions. Here

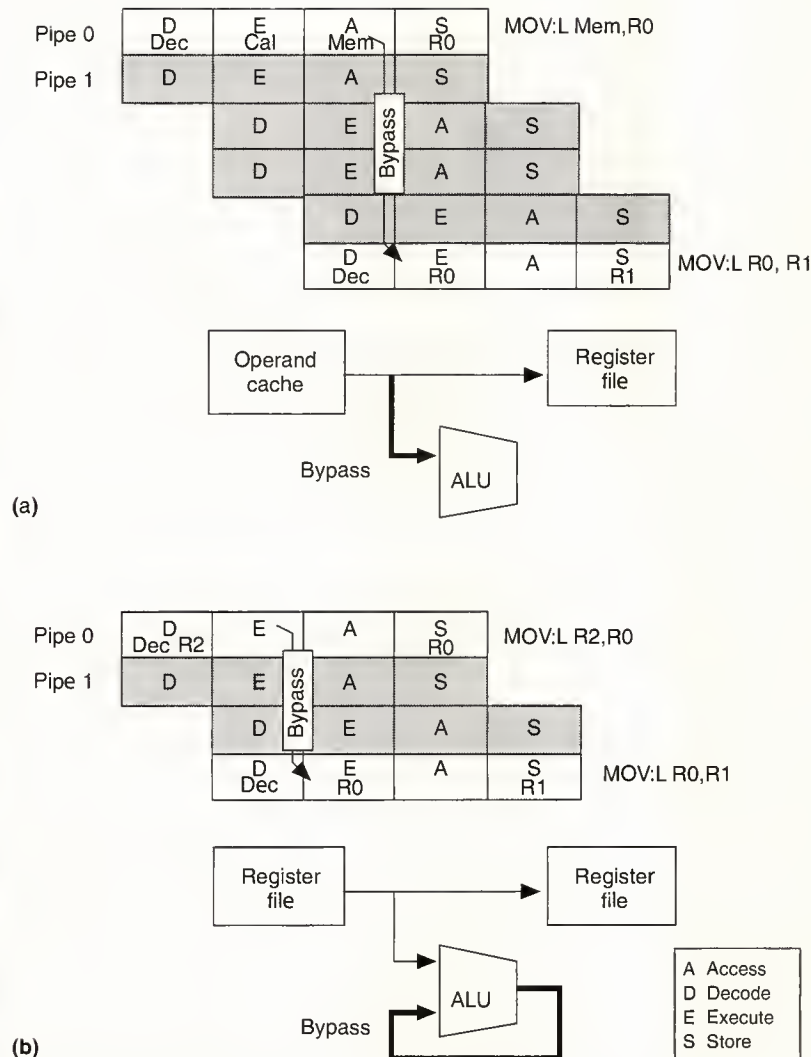


Figure 10. Bypass function: load-use conflict (a); define-use conflict (b).

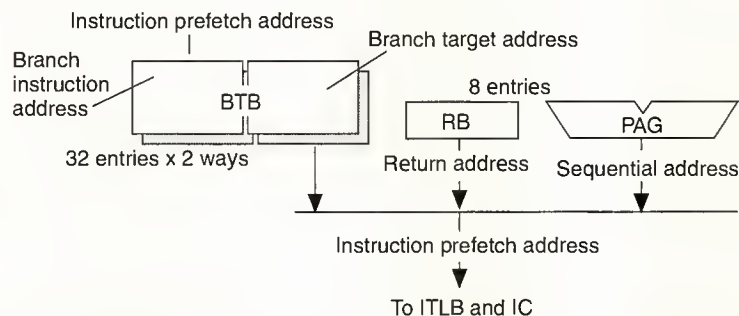


Figure 11. Structure of branch buffers.

tion, the Gmicro/500 features resident buffers dedicated to storing branch information. The branch target buffer, which has a two-way, set-associative structure, stores the addresses of branch instructions and each corresponding branch target address. This enables high-speed execution of branch-always and branch-to-subroutine instructions. The return buffer, which operates as a first-in, last-out memory, caches return addresses to allow high-speed execution of the return-from-subroutine instructions. Figure 11 shows the connection between the branch target buffer, the return buffer, and the prefetch address generator in the instruction unit. The processor selects the outputs from the branch target buffer, the return buffer, and the prefetch address generator. It then uses the selected logical address to access the instruction TLB and the instruction cache.

Zero-cycle branch. Use of the branch target buffer accelerates the execution of branch-always and branch-to-subroutine instructions. Figure 12 shows a zero-cycle branch in the branch-always execution. Figure 12a illustrates the effect on the pipeline when the processor executes a branch operation without the branch target buffer. The chip calculates the branch target address in stage E of the branch-always pipe. This address becomes the next instruction prefetch address (stage I in the next pipe), so the branch-always instruction requires a three-cycle penalty before executing the branch target instruction.

In Figure 12b, the chip effectively executes the branch-always instruction without having a pipe use the branch target buffer, thus allowing a zero-cycle branch. The instruction prefetch address of the branch-always instruction looks up the branch target buffer. It then reads the branch target address from the branch target buffer in case of a branch target buffer hit. The branch target address automatically prefetches the target instruction of the branch-always instruction, and the processor immediately executes the instruction residing at that location.

Fast return from subroutine. Using the return buffer accelerates the execution of the return-from-subroutine instruction. Figure 13a illustrates the effect on the pipe-

line when the chip executes a return operation without the return buffer. The processor pops the return address from the stack in stage A of the return-from-routine first pipe. It uses that address to fetch the target instruction of the return-from-routine through stage E of the last return-from-subroutine pipe. The return-from-subroutine instruction thus requires a five-cycle penalty before it executes the target instruction.

In Figure 13b, use of the return buffer reduces the penalty of the subroutine return from five cycles to two. In stage D of the return-from-subroutine instruction, the chip looks up the return buffer and reads the return address from the return buffer in case of a return-buffer hit. The return address automatically prefetches the return target instruction, and the chip immediately executes the instruction residing at that location.

Effect of branch target buffer and return buffer. Since the Gmicro/500 incorporates the branch target buffer and the return buffer, the compiler tries to make effective use of these resources. In the example of Figure 14 (next page), the compiler minimizes the number of taken conditional branches by introducing zero-cycle unconditional branches. In this way, it reduces the overhead of pipeline hazards caused by taken conditional branches. Figure 14a shows the nonoptimized case where the loop executes in four cycles because of the taken penalty of the conditional branch instruction.

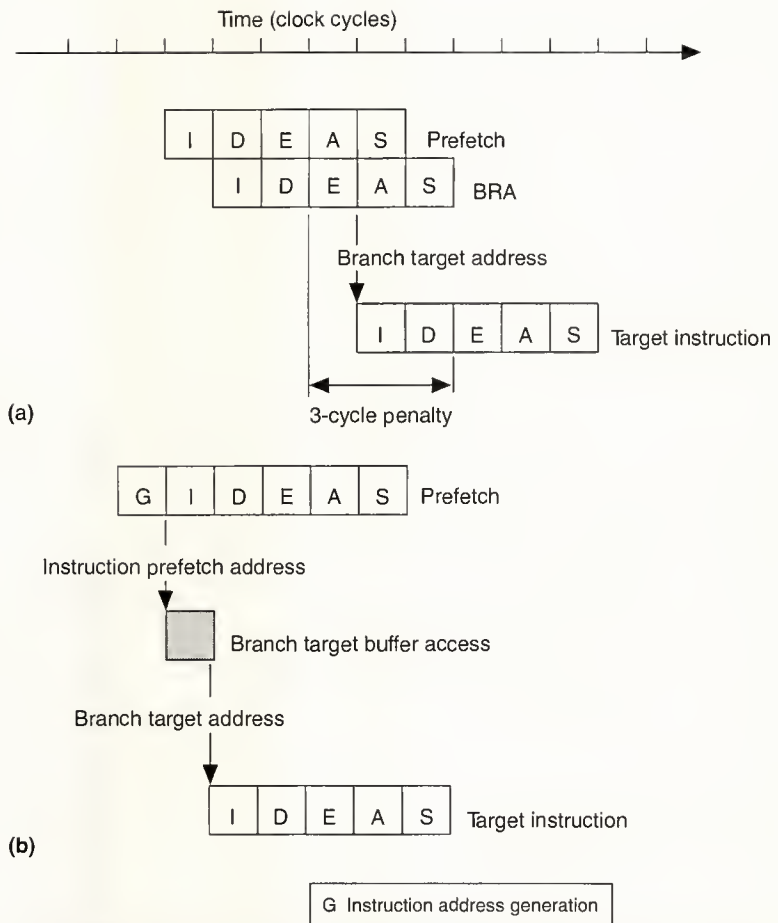


Figure 12. Zero-cycle branch: without branch transfer buffer (a); branch transfer buffer hit (b).

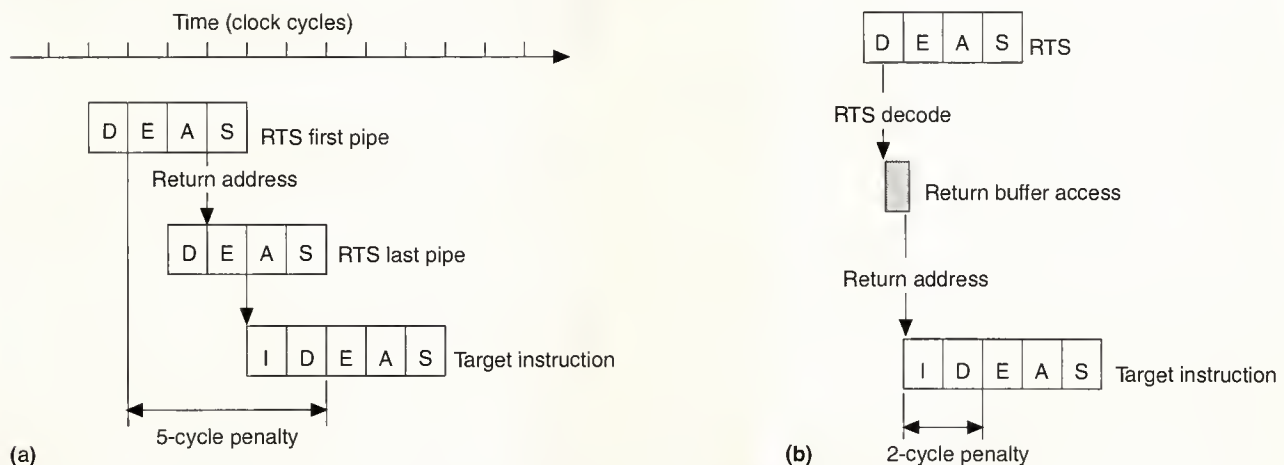


Figure 13. Fast return from subroutine: without return buffer (a); return-buffer hit (b).

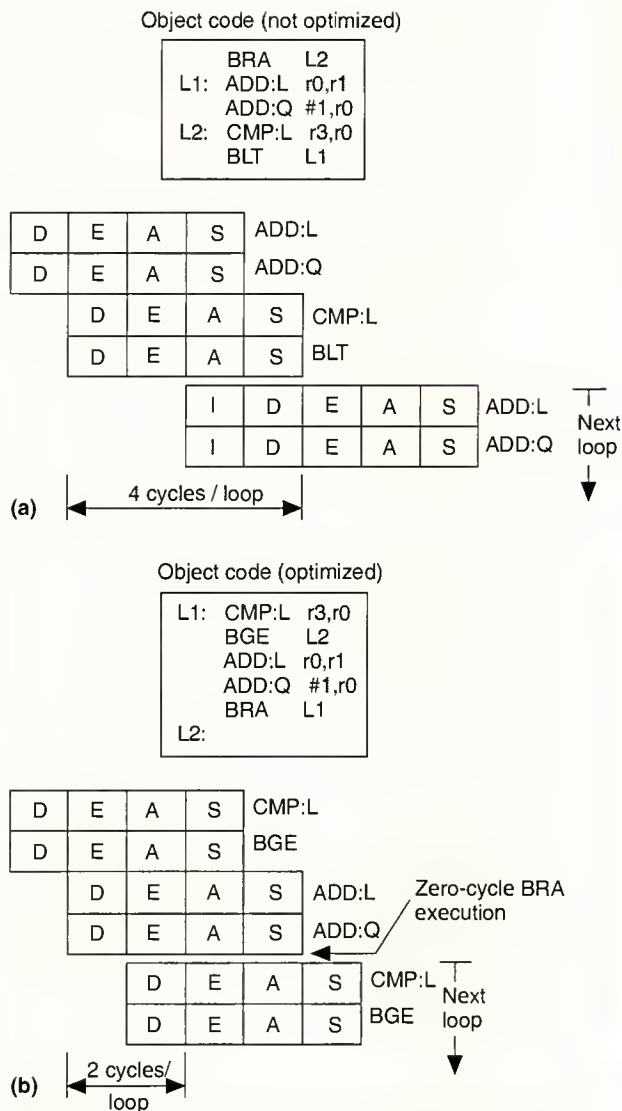


Figure 14. Branch optimization: nonoptimized (a); optimized (b).

Figure 14b demonstrates the optimized case where the loop executes in two cycles. Since the conditional branches are not taken during the loop in Figure 14b, no branch penalty occurs. The chip also uses the branch target buffer for zero-cycle execution of the branch-always instruction. Figure 15 shows the effect of the branch target buffer and the return buffer in the Dhrystone benchmark. Using the branch buffers effectively minimizes the execution time of branch-always, branch-to-subroutine, and return-from-subroutine instructions, reducing the total execution time to 73 percent.

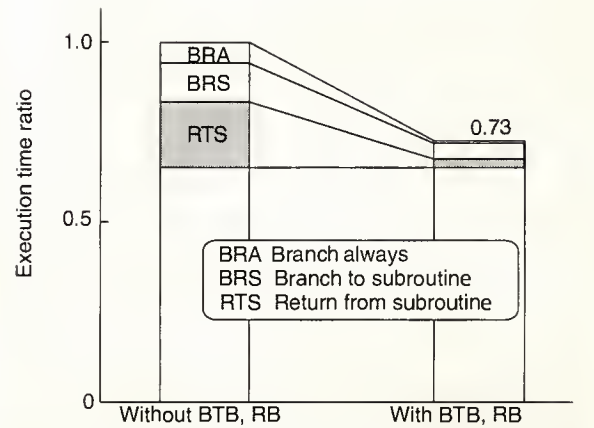


Figure 15. Effect of the branch transfer buffer and the return buffer.

Time	Pipe 0	Pipe 1
72	ADD:Q #imm,r5	
73		MOV:S r5,r0
74	MOV:I #imm:32,r2	
75	SHL:Q #imm,r0	MOV:S r5,r1
76	ADD:L r0,r7	MUL:R r2,r1
77		
78	MOV:S r4,@(r7)	ADD:L r6,r1
79	MOV:S r5,r2	MOV:S r4,@(d:16,r7)
80	SHL:Q #imm,r2	MOV:S r5,@(d:16,r7)
81	ADD:L r1,r2	
82		MOV:S r2,r6
83	MOV:S r5,@(r2)	ADD:Q #imm,r2
84	MOV:L '@(d:16,r6),r1	
85		MOV:S r5,@(r2)
86	ADD:Q #imm,r1	MOV:L '@(r7),r2
87	MOV:S r1,@(d:16,r6)	
	⋮	⋮

Figure 16. Parallel execution of dual pipelines.

Analysis of performance

Using the superscalar structure, branch buffers, and hardware-dependent optimizations for object-codes, the Gmicro/500 achieves high performance in a variety of programs. See Figure 16 for an example of dual-pipeline use. The code sequence used is part of the Dhrystone benchmark. The Gmicro/500 is executing two instructions simultaneously at times 75, 76, 78, 79, and so forth. The figure shows several pipeline hazards that prevent parallel execution. For example, register conflicts occur at times 72-73 and 81-82. The operand cache is conflicted at time 84-85. In this example, the chip executes two instructions simultaneously in 47 percent of cycles during this period despite various pipeline conflicts.

Figure 17 shows the results of dual-pipeline use and performance in Dhrystone Version 1.1 benchmark.¹² The graph shows the execution cycles of a Dhrystone loop. The Gmicro/500 can execute one Dhrystone loop in 285 cycles with the optimized object codes. This performance level is equivalent to 132 MIPS at 66 MHz (1 MIPS = 1,757 Dhrystone loops/s). The processor executes two instructions simultaneously on pipe 0 and pipe 1 in 65 cycles per loop. Using the dual ALUs and barrel shifters in the superscalar integer unit, the chip executes high-level instructions under microprogram control at a rate of 60 cycles per loop. There are 44 wait cycles in the loop caused by an operand interlock, a branch penalty, and so on. Pipe 1 is effectively used in 105 cycles in this Dhrystone loop, achieving a dual-pipeline performance gain of 27 percent (compared to 390 cycles to execute the loop with only one pipe in the chip).

The Gmicro/500 is upward object-code compatible with earlier members of the Gmicro family of processors, so software and other development tools such as the Gmicro/200 compiler can be used without modification. An optimized compiler for the Gmicro/500 based on the Gmicro/200 C compiler is currently under development.¹³ To attain high performance, the optimized compiler adds functions that take full advantage of the Gmicro/500's unique features, such as its superscalar structure and branch buffers.

SYSTEMS USING THE GMICRO/500 ARE under development in such fields as communication switching systems and industrial equipment. Currently, we are evaluating the performance advantage of the superscalar pipeline, branch buffers, and multiprocessing functions under the real circumstances of operating systems and application programs. As we go, we will use the results of these evaluations to develop an even more powerful Gmicro processor. ■

Acknowledgments

We thank Ken Sakamura of the University of Tokyo, as well as Hideo Inayoshi, Tadahiko Nishimukai, and the Gmicro staff members for their helpful suggestions. We also acknowledge all members of the Gmicro/500 development team for their valuable work.

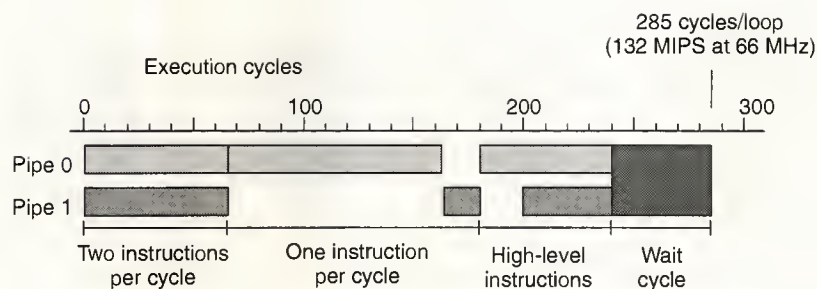


Figure 17. Dhrystone benchmark results.

References

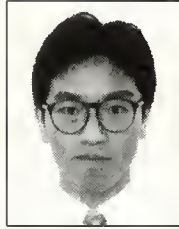
1. D. Dobberpuhl et al., "A 200-MHz 64-bit Dual-Issue CMOS Microprocessor," *ISSCC Digest of Technical Papers*, Feb. 1992, pp. 106-107.
2. F. Abu-Nofal et al., "A Three-Million-Transistor Microprocessor," *ISSCC Digest of Technical Papers*, 1992, pp. 108-109.
3. J. Yetter et al., "A 100-MHz Superscalar PA-RISC CPU/Coprocessor Chip," *Symp. VLSI Circuits Digest of Technical Papers*, 1992, pp. 12-13.
4. J. Crawford et al., "The P5 Microarchitecture," *Fifth Ann. Microprocessor Forum*, 1992, pp. 11-1 to 11-16.
5. K. Sakamura, "Architecture of the TRON VLSI CPU," *IEEE Micro*, Vol. 7, No. 2, Apr. 1987, pp. 17-31.
6. T. Yoshida et al., "The Gmicro/100 32-bit Microprocessor," *IEEE Micro*, Vol. 11, No. 4, Aug. 1991, pp. 20-23, 62-72.
7. H. Inayoshi et al., "Realization of the Gmicro/200," *IEEE Micro*, Vol. 8, No. 2, Apr. 1988, pp. 12-21.
8. T. Kitahara and T. Satoh, "The Gmicro/300 32-Bit Microprocessor," *IEEE Micro*, Vol. 10, No. 3, June 1990, pp. 68-75.
9. T. Yoshida et al., "The Approach to Multiple Instruction Execution in the Gmicro/400 Processor," *Proc. Eighth Int'l Symp. TRON Project*, IEEE Computer Society Press, Los Alamitos, Calif., 1991, pp. 185-195.
10. S. Matsui et al., "Gmicro/500 Microprocessor: Pipeline Structure of Superscalar Architecture," *Proc. Ninth Int'l Symp. TRON Project*, IEEE CS Press, 1992, pp. 56-62.
11. F. Arakawa et al., "A CMOS 50-MHz CISC Superscalar Microprocessor," *Symp. VLSI Circuits Digest of Technical Papers*, 1993, pp. 11-12.
12. R.P. Weicker, "Dhrystone: A Synthetic Systems Programming Benchmark," *Comm. ACM*, Vol. 27, No. 10, Oct. 1984, pp. 1013-1030.
13. Y. Kashiwagi et al., "An Optimizing C Compiler for the Gmicro/500 Microprocessor," *Proc. Ninth Int'l Symp. TRON Project*, 1992, pp. 63-69.



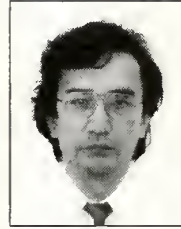
Uchiyama



Arakawa



Narita



Aoki



Kawasaki



Matsui



Yamamoto



Nakagawa



Kudo

Kunio Uchiyama is a senior researcher in Hitachi's Central Research Laboratory, where he holds responsibility for research and development of microprocessors. His interests include microprocessors, cache memories, and design automation tools.

Uchiyama received the BS and MS degrees in information science from the Tokyo Institute of Technology. He is a member of the IEEE.

Fumio Arakawa is a researcher in Hitachi's Central Research Laboratory, where he has been involved in 32-bit microprocessor design.

He received the BS and MS degrees in applied physics from the University of Tokyo.

Susumu Narita is a researcher in the Central Research Laboratory, where he has been involved with 32-bit microprocessor and design tools.

Narita received the BE and ME degrees in electrical engineering from Hokkaido University.

Hirokazu Aoki is a researcher in Hitachi's Central Research Laboratory, where he has been involved with cache memories and 32-bit microprocessor. He graduated from Shizuoka Technical High School.

Ikuya Kawasaki is an engineer in the Semiconductor and Integrated Circuits Division at Hitachi where he has been involved with 32-bit microprocessor. He holds responsibility for development of Gmicro/500.

He received the BS degree in electrical engineering from the University of Tokyo and the MS degree from the University of Washington.

Shigezumi Matsui is an engineer in the Semiconductor and Integrated Circuits Division at Hitachi, where he has been involved in 32-bit microprocessor design.

Matsui received the BE degree in mathematical engineering and instrumentation physics from the University of Tokyo.

Mitsuyoshi Yamamoto is an engineer at Hitachi's Semiconductor and Integrated Circuits Division, where he has been involved with 32-bit microprocessor.

He received the BE and ME degrees in electrical engineering from Nagoya University.

Norio Nakagawa is an engineer at Hitachi's Semiconductor and Integrated Circuits Division, where he has been involved with the 32-bit microprocessor and floating-point coprocessor. Nakagawa received the BS degree in physics from the Tokyo Institute of Technology.

Ikuo Kudo is an engineer in the Semiconductor and Integrated Circuits Division at Hitachi, where he has been involved with microprocessors for watch and automobile applications and with 32-bit microprocessors. Kudo graduated from Kitami Technical High School.

Address questions concerning this article to Kunio Uchiyama at Hitachi, Ltd., Central Research Laboratory, 1-280, Higashi-koigakubo, Kokubunji-shi, Tokyo 185, Japan; uchiyama@crl.hitachi.co.jp.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 153

Medium 154

High 155

IEEE COMPUTER SOCIETY PRESS BOOKS

VLSI ALGORITHMS AND ARCHITECTURES Fundamentals

edited by N. Ranganathan

Addresses introductory and fundamental ideas related to VLSI algorithms and architectures and provides a concise tutorial on the topics. The chapters in this volume introduce basic concepts, present papers on systolic and wavefront arrays, focus on VLSI implementation of data structures, deal with VLSI structures for matrix and algebraic computations, and address important application areas. Overall, the book provides a variety of useful reference material for researchers and students seeking information on the design of hardware algorithms and architectures for VLSI implementation.

Sections: An Overview of VLSI Algorithms and Architectures, Systolic and Wavefront Arrays, Data Structures and Sorting, Matrix and Algebraic Computations, Pattern Matching and Text Retrieval, VLSI Processor Designs.

*304 pages. August 1993. Hardcover. ISBN 0-8186-4392-7.
Catalog # 4392-01 — \$40.00 Members \$32.00*

CODES FOR DETECTING AND CORRECTING UNIDIRECTIONAL ERRORS

edited by Mario Blaum

This collection presents state-of-the-art theory and practice for codes that correct or detect unidirectional errors not covered by most books on error-correcting codes. It features key papers demonstrating the best results in each subject related to unidirectional errors. It is intended for engineers working in computer memory technologies and VLSI design, researchers in the area of error control codes, and graduate students needing access to this cutting-edge material.

Sections: Unidirectional Errors, Codes for Detecting Unidirectional Errors, Codes for Correcting Unidirectional Errors, Codes for Correcting t -Symmetric Errors and Detecting All Unidirectional Errors, Codes for Correcting and Detecting Combinations of Symmetric and Unidirectional Errors, Codes for Detecting and/or Correcting Unidirectional Burst Errors and Byte Errors.

*224 pages. June 1993. Hardcover. ISBN 0-8186-4182-7.
Catalog # 4182-03 — \$44.00 Members \$35.00*

VLSI ALGORITHMS AND ARCHITECTURES Advanced Concepts

edited by N. Ranganathan

Contains many new studies exposing various computationally intensive problems requiring VLSI solutions and addressing advanced concepts as well as VLSI architectures for a broad spectrum of application areas. The chapters discuss important architectural design issues, focus on advanced concepts for systolic arrays and algorithms, and describe special-purpose architectures for a wide range of computationally intensive problems. The papers throughout this volume cover problems from a wide range of application areas that require VLSI solutions.

Sections: VLSI Architecture Design Issues; Advanced Topics in Systolic Arrays; Image, Speech, and Signal Processing; Artificial Intelligence and Computer Vision; Dictionary Machines and Data Compression; Iterative Algorithms.

*312 pages. August 1993. Hardcover. ISBN 0-8186-4402-8.
Catalog # 4402-01 — \$40.00 Members \$32.00*

REAL-TIME SYSTEMS Abstractions, Languages, and Design Methodologies

edited by Krishna M. Kavi

Presents important information on new formalisms, high-level programming languages, and CASE tools that increase the effectiveness of real-time systems. It includes survey articles, specifically written for this tutorial, and a collection of major research papers addressing formalisms, languages, and design methodologies. Its chapters describe some common myths regarding the use of formalisms in software development; detail the use of real-time temporal logic, process algebras, petri nets, and assertional methods; and examine formalisms based on operational semantics, axiomatic logic, and denotational semantics.

Sections: Real-Time Systems; Perspectives, Real-Time Specification and Verification, Real-Time Languages, Design Methodologies for Real-Time Systems.

*672 pages. 1992. Hardcover. ISBN 0-8186-3152-X.
Catalog # 3152-01 — \$70.00 Members \$55.00*

ALSO AVAILABLE !



IEEE COMPUTER SOCIETY
10662 Los Vaqueros Circle
Los Alamitos, CA 90720

Call toll-free 1-800-CS-BOOKS
in CA (714) 821-8380
or FAX (714) 821-4641

VLSI DESIGN 1993

392 pages. January 1993. ISBN 0-1886-3180-5.
Catalog # 3180-02 — \$80.00 Member \$40.00

11TH ANNUAL 1993 VLSI TEST SYMPOSIUM

384 pages. April 1993. ISBN 0-1886-3830-3.
Catalog # 3830-02 — \$70.00 Member \$35.00



The μ VP 64-Bit Vector Coprocessor:

A New Implementation of High-Performance Numerical Computation

Intended for high-speed numerical computations, this coprocessor represents a true vector architecture on a single CMOS chip. It integrates 1.5 million transistors and offers 206-Mflops performance.

Makoto Awaga

Fujitsu Limited

Hiromasa Takahashi

Fujitsu Laboratories Limited

Although supercomputers are used extensively in scientific applications such as fluid dynamics and structural analysis, they are costly to purchase, operate, and maintain. Users seek less costly alternatives that make use of open systems for scientific applications to increase engineering workstation performance. Unfortunately, the superscalar, VLIW, or superpipeline architectures used in popular high-performance engineering workstations have been unable to match supercomputer performance. The μ VP is a new engine for use in conjunction with those microprocessors; it offers the performance of a supercomputer in open-system scientific applications.

The μ VP is a single-chip vector coprocessor developed to meet the needs of high-performance processors in the coming years. It is the world's first supercomputer component implemented onto a single LSI (large-scale integrated) chip developed in CMOS technology. It introduces vector processing technology, a popular method for achieving high performance in the scientific computing community, to the microprocessor world. With 206-Mflops single-precision and 106-Mflops double-precision performance at 50 MHz, the μ VP offers a rate almost equivalent to the performance of typical mini-supercomputers.

Architecture

Most of the high-end scalar processors used in open systems are designed with pipelined architectures. Using new pipelined architectures (superpipeline, superscalar, VLIW) has significantly improved the performance of those scalar processors. However in some scientific applications in which a large amount of low locality data is accessed from a long running program, these scalar processors appear incapable of achieving their advertised performance. One of the main reasons is the physical limitation of the cache memory. In most of the scalar processors performance depends on the cache hit ratio. Therefore, in applications having very low data locality but large volume, cache misses occur at every operand access, penalizing performance due to the need to refresh the cache contents.

We developed a function model of a superscalar processor and simulated this relationship between cache size and cache miss ratio. Our simulation used a model of DAXPY, which is a part of the Linpac benchmark program of Jack J. Dongarra at the University of Tennessee.¹ We took the Clean-up loop out of the source and put it in the two-dimensional loop shown in Figure 1.

N is assumed to be a multiple of 4. We checked the cache miss ratio, changing the matrix size of

the source data (value of N). Figure 2 indicates the result of our simulation. We configured the superscalar processor to have the following:

- 1) a 66-MHz clock frequency (a 15-ns cycle);
- 2) a capability for issuing up to three instructions every cycle: INT+INT+FP or INT+FP+FP (INT indicates integer operations and FP indicates floating-point operations);
- 3) a load/store operation between the data cache and registers of one operand each cycle;
- 4) four-cycle latencies for FPMUL and FPADD, and two-cycle latency for FPLOAD;
- 5) separate instruction and data caches (we configured the data cache as either 16-Kbyte entries, direct mapped, and a 32-byte line size or 32-Kbyte entries, two-way associativity, and a 32-byte line size);
- 6) a 10-cycle (150-ns) cache miss penalty; and
- 7) a 64-entry, 32 sets \times two-way set associative, 4-Kbyte-page data TLB (translation look-aside buffer) with a 30-cycle (450-ns) miss penalty.

We converted the source program based on the DAXPY Clean-up loop into the codes in our simulation according to the specified configuration in Figure 3.

Although our simulation achieved a peak performance of 132 Mflops at 66 MHz (two floating-point operations can be executed at a time), once the matrix size exceeds 100 \times 100 the 10 percent cache miss causes a significant decrease in sustained performance. We attempted four iterations of loop unrolling to minimize the overhead of conditional branches. However, the sustained performance remained at less than 20 percent of peak when the matrix size was greater than 100 \times 100. This simulation assumed a TLB hit of 100 percent and therefore did not include TLB penalties.

A vector register machine performs all vector operations except loads and stores on data stored in vector registers. Unlike cache memory architectures, software con-

```

DO 20 I = 1,N
  DO 10 J = 1,N,4
    DY(I, J) = DY(I, J)+DA*DX(I, J)
    DY(I, J+1) = DY(I, J+1)+DA*DX(I, J+1)
    DY(I, J+2) = DY(I, J+2)+DA*DX(I, J+2)
    DY(I, J+3) = DY(I, J+3)+DA*DX(I, J+3)
  10 CONTINUE
20 CONTINUE
  
```

CLEAN-UP LOOP
part of DAXPY

Figure 1. Two-dimensional loop.

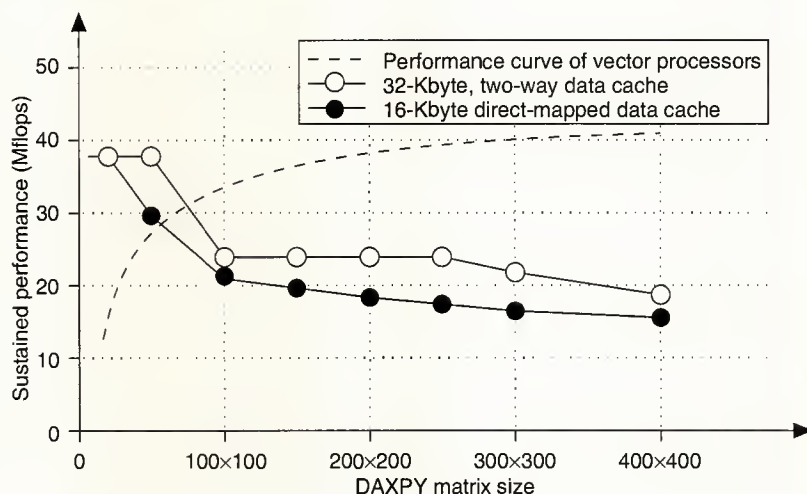


Figure 2. Simulation results of a superscalar processor's sustained performance.

	— Load/Store —	— FP mul/add —	— Integer operation —	
Loop:	ld.d Y[i+3]	fmul.d Z[i]	add p X[i]	(1)
	ld.d X[i]	fmul.d Z[i+1]	add p X[i+1]	(2)
	ld.d X[i+1]	fmul.d Z[i+2]	add p X[i+2]	(3)
	ld.d X[i+2]	fmul.d Z[i+3]	add p X[i+3]	(4)
	ld.d X[i+3]	fadd.d Z[i]	add p Z[i]	(5)
	st.d Z[i]	fadd.d Z[i+1]	add p Z[i+1]	(6)
	st.d Z[i+1]	fadd.d Z[i+2]	add p Z[i+2]	(7)
	st.d Z[i+2]	fadd.d Z[i+3]	add p Z[i+3]	(8)
	st.d Z[i+3]		add p Y[i]	(9)
	ld.d Y[i]*		add p Y[i+1]	(10)
	ld.d Y[i+1]*		add p Y[i+2]	(11)
		sub cc	add p Y[i+3]	(12)
	ld.d Y[i+2]*		brc loop	(13)

Figure 3. Converted codes for simulation. (* indicates software pipelining used.)

Table 1. Product lineup.

Item	Description
Part numbers	MB92831-33/-45/-50
Operation frequency	33/45/50 MHz
External bus configuration	
Data bus	64 bits
Address bus	32 bits
Bus bandwidth	264/360/400 Mbytes/s
Internal registers	
Vector (VR)	8 Kbytes
Mask (VMR)	64 bytes
Scalar (VSR)	128 bytes
Command buffer (VCB)	1 Kbyte
Address translation (TR)	512 bytes
Control	11
Internal execution pipelines	
Add	64-bit Adderx1 or 32-bit Adderx2
MUL	64-bit Multiplierx1 or 32-bit Multiplierx2
DIV	64-bit Dividerx1
Graph	Graphic operation
Mask	Mask operation
Applicable data format	Single-/double-precision floating-point (IEEE Std. 745)
Command format	32-bit integer/logical/pixel
No. of commands/type	32-bit, fixed
	141 vector operation
	57 scalar operation
	9 general control
Peak performance	
Single-precision	136/186/206 Mflops
Double-precision	70/96/106 Mflops
Debugging support	IEEE Std. 1149.1, JTAG
Supply voltage	3.3V
Power dissipation	3.0/4.5/5.0 watts
Package	Ceramic SQFP-256 (0.5-mm lead pitch)

- no overhead for branch condition control,
- low cost of latency to main memory access, and
- vector instructions that replace entire loops.

Design considerations

In the scientific computation arena, where very large active data sets are accessed with low locality, vector supercomputers enjoy great popularity due to the useful properties just discussed. However, many users normally share these vector supercomputers, and a huge amount of initial investment and running costs is required.

To take advantage of vector operations in personal systems and drastically improve the numerical computation performance, we implemented a vector architecture onto a single CMOS LSI chip. In this implementation we tried to

- reduce the start-up overhead caused by the pipeline latency,
- make a host processor interface universal, so that it can be attached and used with various types of microprocessors,
- build a memory interface using conventional DRAMs and offer enough speed to synchronize to internal operations,
- put all the necessary execution pipelines and reasonable size of vector registers on a chip for which reasonable production yield can be expected, and
- mount the chip in a reasonably small package so it can be used for open-system products and/or cost sensitive board-level products.

controls the contents of vector registers, thus avoiding performance constrictions caused by cache misses. In comparison, vector processors tend to exhibit performance curves as shown by the dashed line in Figure 2. Due to the start-up overhead of vector processors, superscalar processors achieve better performance when the matrix size is less than 50×50. However, once the matrix size exceeds 50×50, vector processor performance increases, and the increase is maintained regardless of the matrix size.

Since we designed the μ VP based on this vector register machine architecture, it provides several important properties of vector operations:²

Product overview

Table 1 lists the main features of the μ VP, which integrates approximately 1.5 million transistors into a 15.99×15.99-mm CMOS chip. Figures 4 and 5 show a die photo and the packaged μ VP. The μ VP contains three separated execution pipelines: multiplier, adder/graphic, and divider/mask. While these three execution pipelines work in parallel, the coprocessor transfers an operand between internal register files and external memory through the load/store pipeline. The μ VP achieves the peak performances listed in Table 1 when these three execution pipelines work simultaneously at 50-MHz operation.

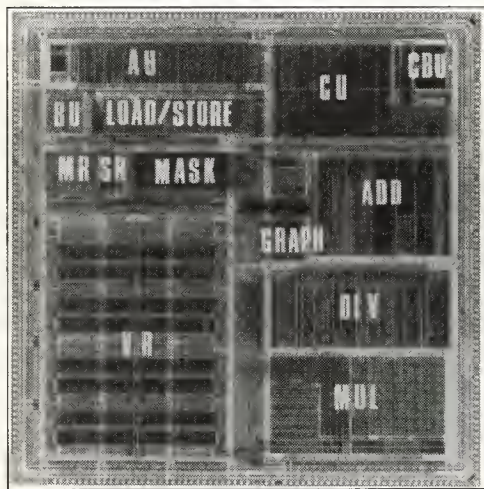


Figure 4. Die photo.



Figure 5. μ VP package appearance.

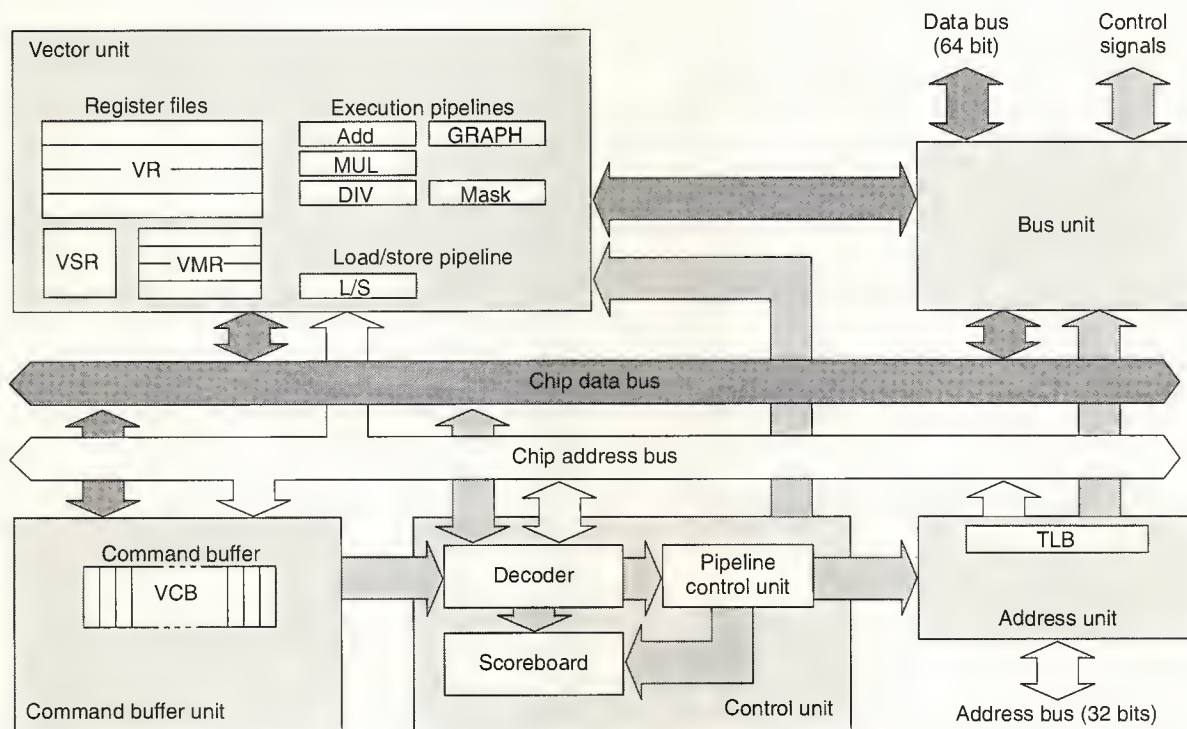


Figure 6. μ VP block diagram.

Figure 6's block diagram shows that the μ VP contains five units: vector, command buffer, control, address, and bus. Among them, the vector unit is the main engine that performs all the calculations (see Figure 7 for its block diagram).

This unit consists of various execution pipelines, a load/store pipeline, and register files.

Execution pipelines. To handle numerical calculations, the μ VP contains multiplier (MUL), adder (Add), divider (DIV),

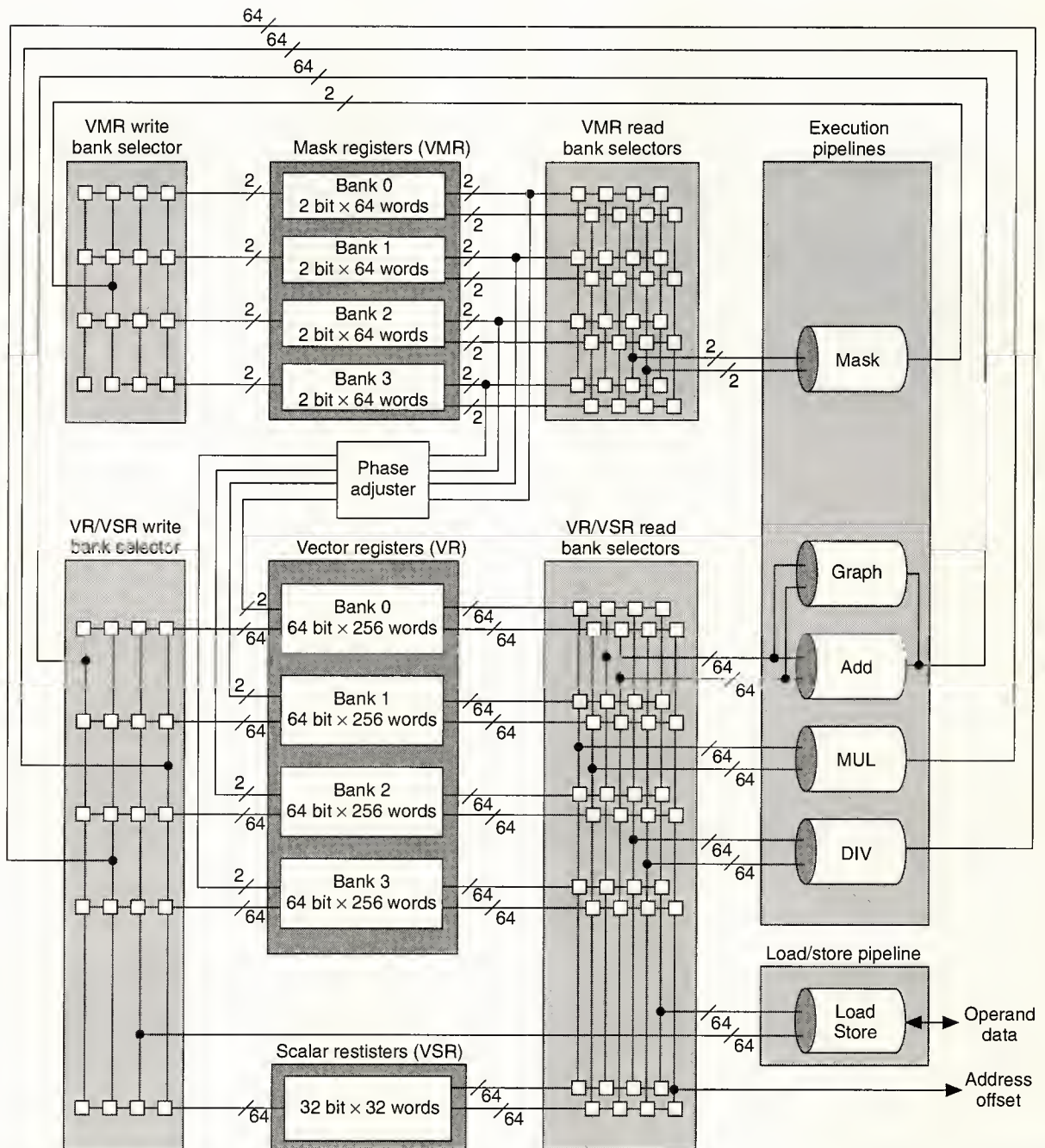


Figure 7. Vector unit block diagram.

graphic operation (Graph), and mask operation (Mask) pipelines. All five pipelines work independently. To work simultaneously, each pipeline has a pair of 64-bit-wide data loading paths and one 64-bit data storing path to register files. How-

ever, since Add and Graph share the same data paths, one of them is automatically selected according to the type of commands. Since DIV and Mask are assigned to the same start-up phase timing, these two do not work simultaneously either.

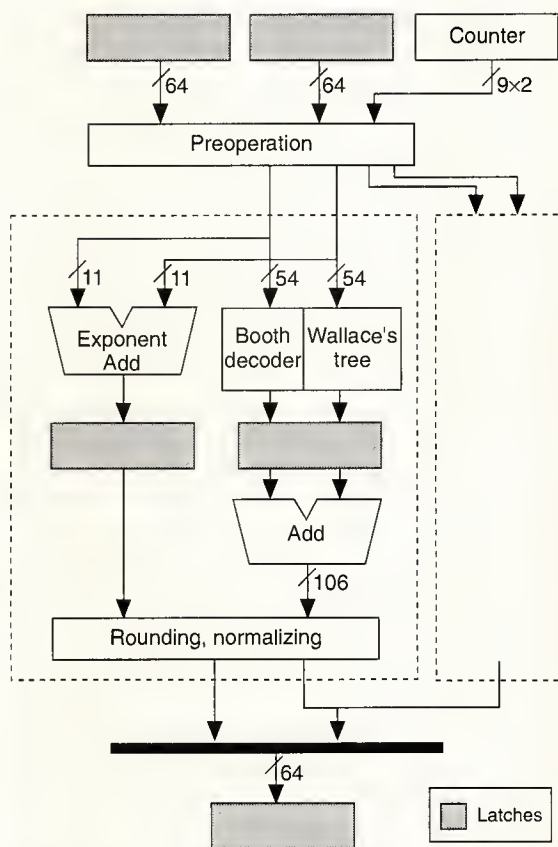


Figure 8. MUL block diagram.

To minimize the start-up overhead, MUL, Add, Graph, and Mask are formed by four stages (source data reads, execution/first half, execution/second half, and destination store). One calculation takes four clock cycles. Since they all work in a pipelined manner, at every cycle a new pair of source data is fed into a pipeline, and at the same time a new destination datum is stored back to the register file. So one calculation completes each cycle. Figures 8 and 9 show the block diagrams of MUL and Add. Both MUL and Add have a single-precision and a double-precision execution plane and a separate single-precision-only execution plane; two single-precision calculations execute at the same time. Therefore at 50 MHz, both of them perform at 50 Mflops for double-precision calculations and 100 Mflops for single-precision calculations. DIV consists of two divider planes, both implemented using iterative radix-16, nonrestoring division.³ Each of those divider planes offers 1/16th the performance of MUL and Add, regardless of the data type; so DIV's total performance is 6.25 Mflops at 50 MHz.

When these pipelines work simultaneously, the μ VP shows

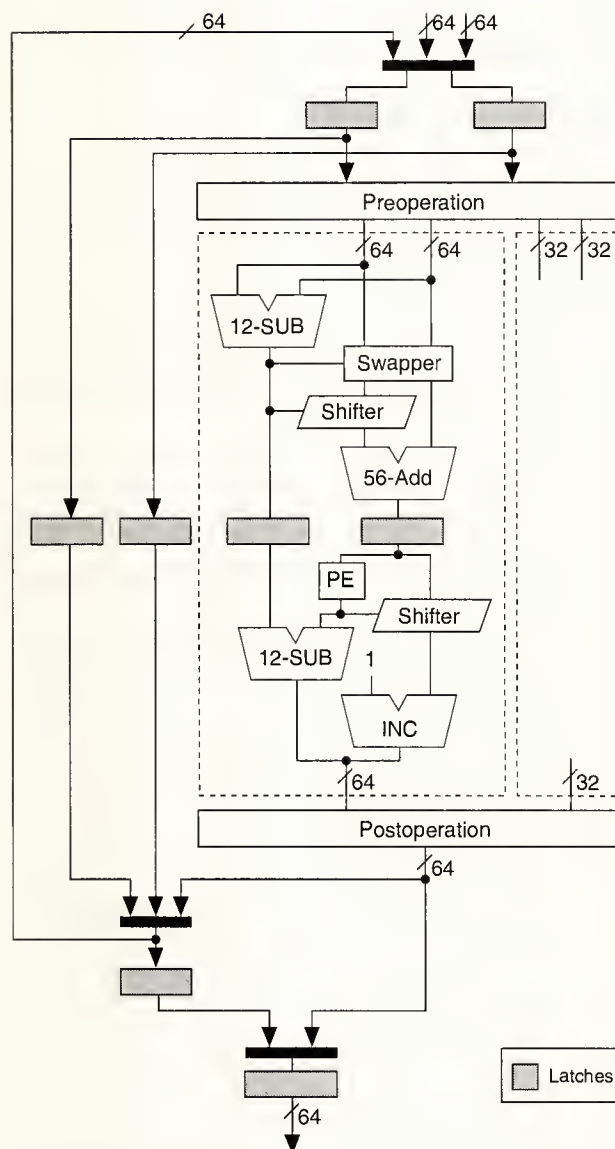


Figure 9. Add block diagram.

peak performance. When both MUL and Add execute single-precision calculations, while DIV works, this peak performance is an addition of those execution pipelines' performance, which is 206 Mflops at 50 MHz.

Load/store pipeline. The 64-bit-wide load/store pipeline (L/S) transfers data between internal register files and external memory. It simultaneously transfers one 64-bit datum or two 32-bit data every clock cycle to the internal pipeline executions.

Vector register. The vector register (VR) contains 8 Kbytes

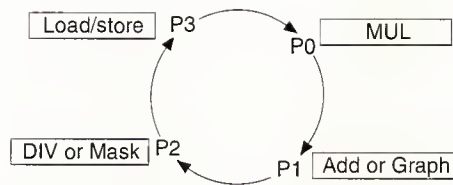


Figure 10. Scheduling of pipeline start timing.

of source and destination vector operands as well as the address offset information for indirect-addressing memory access. The VR consists of three-port SRAMs, so that two read operations and one write operation can take place at the same time. The entire four-way bank organization register file can schedule up to four pipelines to connect to those four VR banks, individually, each clock cycle. Thus, data transfers between VR and pipelines efficiently and simultaneously.

The VR is automatically divided into 8, 16, 32, or 64 partitions (VR0 to VR63), depending on the definition of vector length. The element unit of the VR is 64 bits long. One word of double-precision data or two words of single-precision data can be stored in each element unit. When the VR is divided into eight partitions making each partition 1 Kbyte (64 bits \times 128 words), each partition can hold 128 elements. This means that the maximum applicable vector length for the μ VP is 128 in double-precision or 256 in single-precision calculations.

Before the design of the μ VP started, we considered the expected process yield and decided that the chip size should be smaller than 16 mm \times 16 mm. Although the bigger VR size is always better for vector operations, we enforced this upper limit to the chip. When we developed the floor plan, we considered a rough proportion of the area to be occupied by all other logic parts and such factors of the VR as the cell size, organization, and reasonable capacity. We set the VR size at 8 Kbytes.

Scalar registers. The scalar registers (VSR0 to VSR31) are general-purpose data registers. They contain scalar operands and various parameters to be used in μ VP operations. Typical operations involve a load/store start address, vector stride (a distance between vector elements located next to each other in memory), and so on. Like the VR, the VSR element is also a three-port SRAM. The scalar registers are directly mapped to a host CPU's address field, so that the host CPU can access any of these registers directly.

Mask register. The Mask register (VMR) contains 64 bytes of register file. To vectorize conditional branch operations, this register file stores mask information for the correlated vector operand. Like the VR, the VMR is formed in a four-way bank structure. The element of the VMR is also a three-port SRAM, and the entire register file is divided into 2, 4, 8, or 16 partitions, according to the definition of vector length.

The functional mechanism

To perform vector calculations in high speed simultaneously, we invented a state machine control mechanism called the Bank Slot Phase System.⁴ This phase control system realizes completely synchronized parallel operations of various pipeline executions, data transfers between register files and execution pipelines, and data transfers between register files and external memory. All the operations handled by the μ VP are driven by a Bank Slot Phase clock, which repeats four phases: P1, P2, P3, and P4. The start-up timing of each execution pipeline or load/store pipeline is assigned to one of those clock phases, as shown in Figure 10.

The Bank Slot Phase clock maintains strict control over all data transfer operations between each execution pipeline and the register files (VR, VSR, or VMR) to avoid access conflicts to the same VR bank from multiple pipelines. To control all these timings by hardware, we adopted data distribution management control circuits called bank selectors. Two sets of bank selectors control data loading (transfer from register files to pipelines), while one set controls data storing (transfers from pipelines to register files). Each port of bank selectors is 64-bits wide.

Read bank selectors transfer two sets of 64-bit-wide source data at every clock from register files to all three execution pipelines. The write bank selector transfers 64-bit-wide destination data from all execution pipelines to register files. At the same time the selectors transfer 64-bit-wide data between the load/store pipeline and the register files. Up to seven sets of 64-bit-wide data can transfer from register files to pipelines through read bank selectors, or four sets of 64-bit-wide data can transfer in the opposite way through the write bank selector at every clock cycle. Thus, the bandwidth of those bank selectors are 2,800 Mbytes/s for reads and 1,600 Mbytes/s for writes at 50 MHz.

These bank selectors smoothly switch control for the data transfer between register files and vector pipelines (execution pipelines and load/store pipeline). Table 2 lists the data path assignments at each clock phase controlled by these bank selectors.

Since the selectors are organized by shuffle circuits consisting of multiple tristate drivers (see Figure 11, on p. 32), the physical area of bank selectors is reduced to approximately one fourth of that formed by conventional AND-OR gate arrays.

The efficiency of simultaneous vector executions depends on such conditions as a register conflict or a pipeline hazard. (A conflict occurs when multiple pipelines access the same register at the same time. A hazard occurs when subsequent instructions attempt to use a pipeline already in use.) The program shown in Figure 12a on page 33, for example, does not have conditions that prevent execution pipelines from working in the most efficient way. So in this case, the bank selector executes all four instructions simultaneously, as shown

Table 2. Data transfer path assignment at each clock phase. Screened areas are undefined.

Register files		Bank selectors	Phase number of Bank Slot Phase Clock			
			P0	P1	P2	P3
Vector register (VR)	Bank 0	Read #0	MUL	Add/Graph	DIV*	Store
		Read #1				Address offset **
		Write	Add/	DIV*	Load	MUL
	Bank 1	Read #0	Store	MUL	Add/Graph	DIV *
		Read #1	Address offset**			
		Write	MUL	Add/	DIV*	Load
	Bank 2	Read #0	DIV*	Store	MUL	Add/Graph
		Read #1		Address offset**		
		Write	Load	MUL	Add/	DIV*
	Bank 3	Read #0	Add/Graph	DIV*	Store	MUL
		Read #1			Address offset **	
		Write	DIV*	Load	MUL	Add/
Mask register (VMR)	Bank 0	Read #0			Mask*	
		Read #1				
		Write		Mask*		
	Bank 1	Read #0				Mask*
		Read #1				
		Write			Mask*	
	Bank 2	Read #0	Mask*			
		Read #1				
		Write				Mask*
	Bank 3	Read #0		Mask*		
		Read #1				
		Write	Mask*			

* DIV and Mask do not work simultaneously.

** Applied when indirect addressing mode in use.

in the time sequence chart of Figure 13a on p. 33.

However, in case of a program as shown in Figure 12b, the destination register of the prior vector multiply instruction is referred to as a source operand by the following vector add instruction, and a register conflict occurs between these two instructions. In this case until at least one element is fed back to VR8 by the VMULD instruction, the following VADDD instruction will not be started. But once the products are started after being written back to VR8, all the subsequent operations of VADDD and VMULD can execute simultaneously. Figure 13b shows a time sequence of this operation.

In case of a program similar to the one shown in Figure 12c, instructions to use the same MUL pipeline are put next to each other. Since there is only one MUL in the vector unit, two multiply instructions cannot execute at the same time. Therefore, the following VMULD instruction (2) is not started until the prior VMULD instruction (1) completes. Figure 13c shows a time sequence of this operation.

These conflict conditions are automatically detected by a scoreboard unit residing in the control unit. Once a conflict is detected, the coprocessor automatically postpones execution of the following instruction, which caused the conflict, until that condition clears. Programmers don't need to worry about the overrun and/or pipeline scheduling. Also, as shown in Figure 13b, once the conflict condition clears, all the subsequent operations proceed simultaneously. So the longer the vector length, the less the conflict overhead becomes relative.

Bus configuration

To make handling of the package easy, we tried to keep the package as small as possible. The most useful way to do so is to minimize either pin count or lead pitch. Since the μ VP is a 64-bit vector coprocessor to be used with a 32-bit microprocessor, the data bus and address bus need to be 64-bits wide and 32-bits wide. This means that one set of the load/store

continued on p. 34

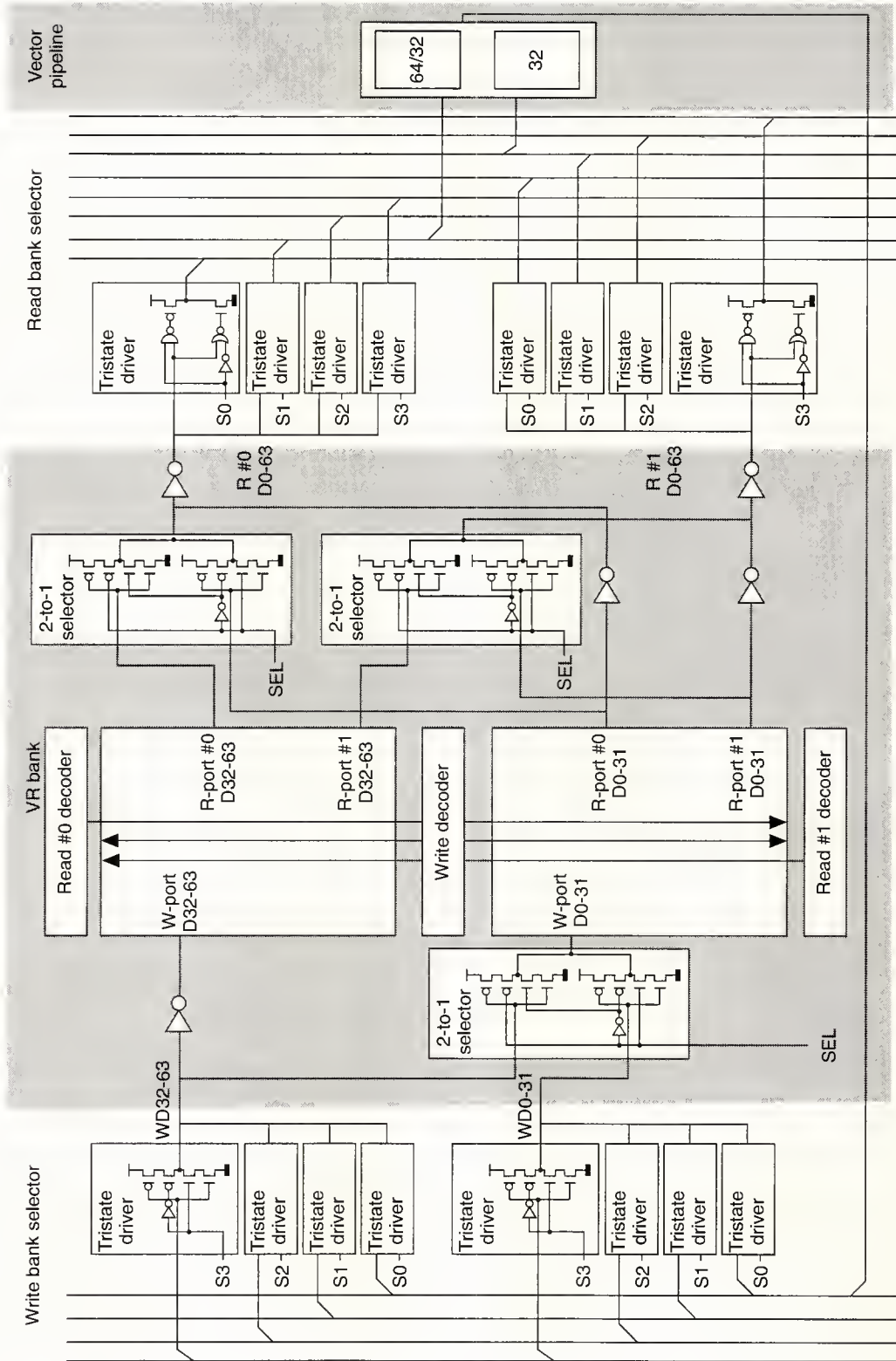


Figure 11. Bank selector circuits.

			; n : vector length (n≤128 in D.P. floating point operand)
vmuld	vr0, vr4, vr8	; D.P. vector multiply	VR0(1:n)*VR4(1:n)=VR8(1:n)
vadd	vr12, vr16, vr20	; D.P. vector add	VR12(1:n)+VR16(1:n)=VR20(1:n)
vdvd	vr24, vr28, vr30	; D.P. vector divide	VR28(1:n)/VR24(1:n)=VR30(1:n)
vst64	vcr36, vsr0, vsr1	; D.P. vector operand store	
(a)			
vmuld	vr0, vr4, vr8	; (1) D.P. vector multiply	
vadd	vr8, vr12, vr16	; (2) D.P. vector add	[start when VR8(1) (first element of above (1) vector multiply) will be ready]
vst64	vr16, vsr0, vsr1	; D.P. vector operand store	[start when VR16(1) (first element of above (2) vector add) will be ready]
(b)			
vmuld	vr0, vr4, vr8	; (1) D.P. vector multiply	
vmuld	vr8, vr12, vr16	; (2) D.P. vector multiply	[start when above (1) vector multiply will all complete]
vst64	vr16, vsr0, vsr1	; D.P. vector operand store	[start when VR16(1) (first element of above (2) vector multiply) will be ready]
(c)			

Figure 12. Programming models: to get the best performance (a), cause a register conflict (b), and cause a pipeline hazard (c).

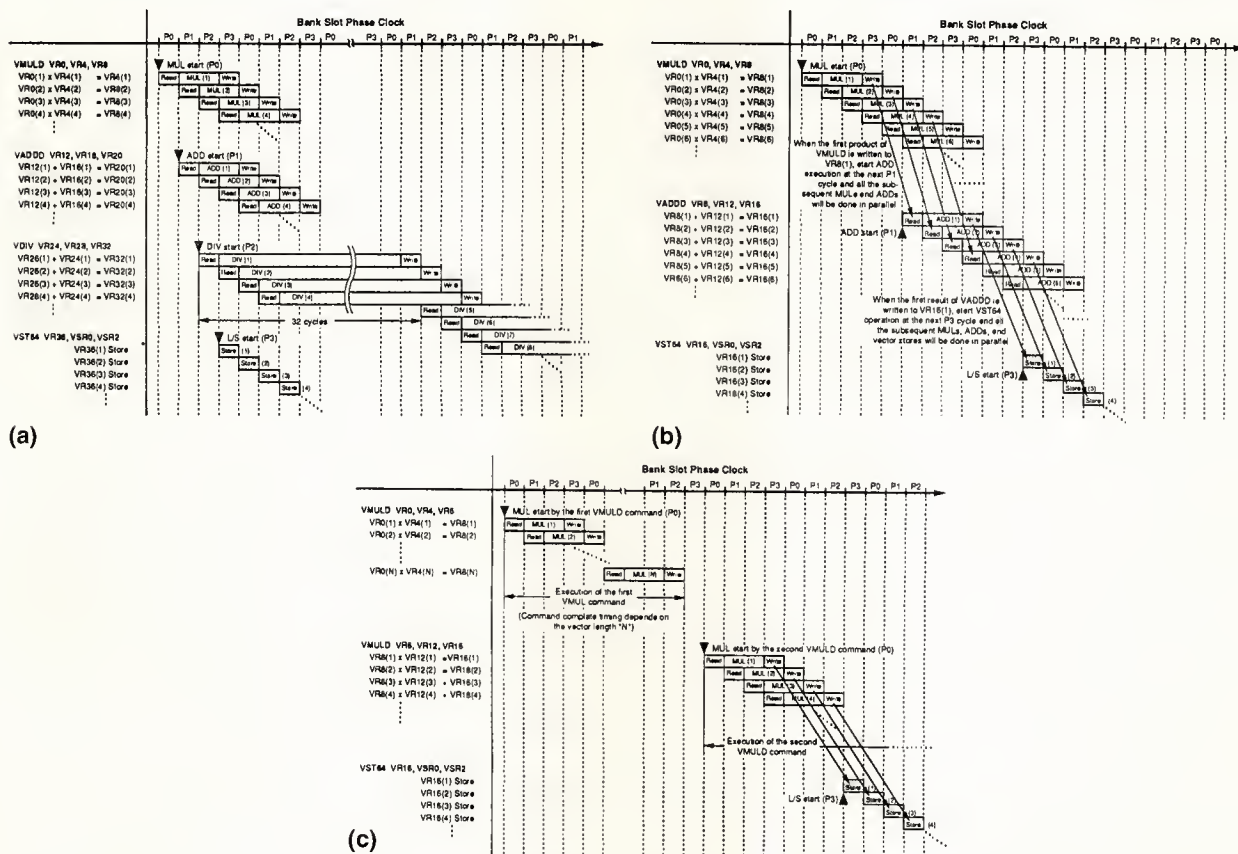


Figure 13. Time sequence charts: maximum efficiency (a), register conflict (b), and pipeline hazard (c).

```

C*
C*  COORDINATE TRANSFORMATION
C*  (OBJECT → WORLD)
C*
C*  SOURCE      : X1(N)=SP(1,N), Y1(N)=
                  SP(2,N), Z1(N)=SP(3,N)
C*  DESTINATION : X2(N)=DP(1,N), Y2(N)=
                  DP(2,N), Z2(N)=DP(3,N)
C*
C*****
C
C  INTEGER*      4 N
C  REAL*4        SP(3,N),DP(3,N)
C  REAL*4        CD(4)
C
C  DO 120 I=1,N
C  DO 110 J=1,3
C  CD(J)=0.0
C  DO 100 K=1,3
C  CD(J)=CD(J)+SP(K,N)*CX(K, J)
100  CONTINUE
C  CD(J)=CD(J)+CX(4, J)
110  CONTINUE
C  DP(1,N)=CD(1)
C  DP(2,N)=CD(2)
C  DP(3,N)=CD(3)
120  CONTINUE

```

Figure 14. Coordinate transformation Fortran source program.

pipeline requires almost 100 pins to communicate with external memories. Considering this point, we equipped just one load/store pipeline, although there are three parallel execution pipelines (MUL, Add/Graph, and DIV/Mask). We kept the total external signal pin count within 130. The chip is mounted in a 1.5x1.5-in., 256 SQFP (shrink quad flat pack), which made it possible to use the μ VP in such cost-sensitive applications as add-on accelerator board products.

A universal host interface makes it possible to use the μ VP with various types of host CPUs. All the internal registers except VR and VMR are address mapped in the host CPU's address space. By issuing the appropriate address and data, the host CPU can access any of these registers while the μ VP is in the idle state.

Vector instructions that access memory have a known access pattern. If the vector's elements are all adjacent, fetching the vector from a set of heavily interleaved memory banks works well.⁵ However, if the memory consists of too many banks, we cannot achieve good performance in short vector length operations due to a high memory access latency. Since

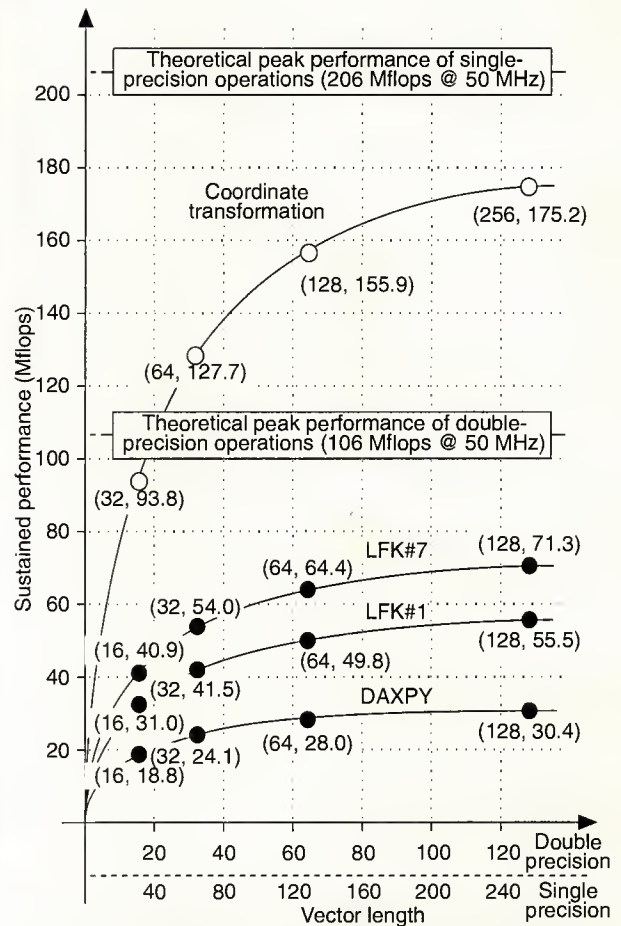


Figure 15. μ VP performance simulation results. Open dots indicate single-precision operations and closed dots, double-precision.

the μ VP is a CMOS single chip, we cannot organize as huge an external memory system as that used in large supercomputers.

As a result, we decided to use a four-way interleaved memory. With consecutive element order, the memory access latency is four cycles. When the μ VP works at 50 MHz and each element access can be completed within 80 ns, no wait state occurs and all the vector operations take place fully synchronized. Even if the interleaved memory is organized by relatively slow DRAMs, this access speed is possible in high-speed page mode.

Command set

The μ VP supports 207 commands. They include 141 vector operation commands, 57 scalar operation commands, and 9 general control commands such as branches and register-to-register internal data transfers. All commands are 32-bits

long and fixed; they contain 10 bits of opcode field and 22 bits of register direction field. We studied our company's command set for vector supercomputers and tried to implement similar μ VP vector operation commands that are optimized to scientific calculations. Those vector operation commands support such complex operations as max/min value find and vector sum to improve the efficiency of vectorization. Scalar operation commands are separately supported from vector operation commands, though scalar operation commands are functionally identical to the vector operation commands with the vector length setting of 1. This approach executes scalar operations simultaneously while vector operations are taking place, without changing the vector length setting.

Program coding example

Figure 14, contains a coding sample of a Fortran source program for coordinate transformation. The μ VP applicable vector length is up to 256 in single-precision calculations. If the element number in the source program is bigger than 256, all data elements must be divided into numbers of segments that the μ VP can manage in bulk vector operation. For each segment, the same set of vector operation is repeated. Although repetition control is required, this overhead is small enough to ignore in comparison to the volume of the vector operation.

Performance evaluation

We simulated the coordinate transformation program sample in our μ VP model. We also used the DAXPY Clean-up loop (a single loop) in the cache miss ratio simulation of a superscalar processor. Other than these two programs, we picked up No. 1 and No. 7 loops from the Livermore Fortran Kernel (LFK), which is a test suite produced by Lawrence Livermore National Laboratory. These three additional simulated benchmark loops check sustained performance in double-precision floating-point operations.

Figure 15 shows the simulation results of these loops. In this figure the three curves marked with dark dots show the results of the double-precision operation of LFK#7, LFK#1, and DAXPY. The curve marked with white dots shows the result of the single-precision operation of the coordinate transformation program. Sustained single-precision performance of this coordinate transformation program is 175.2 Mflops at 50 MHz. The sustained performance is about 80 percent of the theoretical peak performance of 206 Mflops. Since the proportion of internal MUL and add operations is relatively high toward the load/store operations in this coordinate transformation, all the pipeline operations are well synchronized, and the sustained performance comes close to the theoretical peak.

We achieved a sustained performance of 30.4 Mflops, 55.5 Mflops, and 71.3 Mflops at 50 MHz for the DAXPY, LFK#1, and LFK#7 benchmarks. In DAXPY the vector load/store command appears three times as many as the vector MUL or

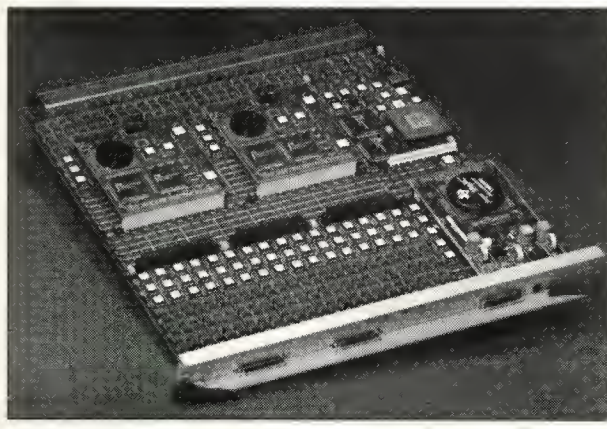


Figure 16. VPE application board.

vector add command. Since the μ VP has only one load/store pipeline, these three load/store commands do not execute in parallel. Vector MUL and vector add executions overlap with one load command execution. But while the other load command and store command execute, all the internal execution pipelines stay in the idle state. This accounts for the sustained DAXPY performance being about 30 percent of the peak performance (106 Mflops at 50 MHz).

As the result of all these simulations, we confirmed that the sustained performance curves of the μ VP follow the characteristics of vector operation shown in Figure 1. As the vector length becomes longer, the sustained performance reaches a certain peak level. There is no such performance drop, as shown in the simulation of a superscalar processor caused by a cache miss. Also, from these results we confirmed that the μ VP provides a quick upward performance curve because of the short pipeline latency. At vector length 16, the sustained performance already reaches 50 percent of the peak of each curve.

We also measured the sustained performance of a real system using the μ VP. This measurement was done on a VPE (vector processor element) board for a modular, massively parallel processing system called CS-2 developed by Meiko. This system was introduced at Supercomputing 1992. Lawrence Livermore National Laboratory plans to install a 256-vector processor element version of this system this year.

Figure 16 shows this VPE board with a couple of μ VPs placed under the control of a SuperSparc chip. Those three processors share a memory system with 128 Mbytes of external DRAM in 16 independent banks. We measured this performance at 33 MHz of operation frequency for three benchmark test programs: DAXPY, DDOT, and DGEMM (matrix multiply). Table 3, next page, lists the results of this evaluation.

Two μ VPs work simultaneously on this board, so the performance result appears to be almost double the single-chip


Table 3. VPE simulation results (Mflops).

Benchmark	At 45 MHz	At 50 MHz
DAXPY ($N = 8,192$)	53.7	59.6
DDOT ($N = 8,192$)	70.6	78.4
DGEMM ($N = 1,000 \times 1,000$)	129.3	143.5

simulation result, according to the DAXPY result comparison. But because of such overhead factors as μ VP start-up from SuperSparc, periodic DRAM refresh cycles, internal repetition control of the μ VP program, and so on, the real result per chip is a little less than the simulation result.

Other than Meiko's CS-2, various kinds of application systems of the μ VP are now available or under development. Currently, the most popular applications of the μ VP are standard bus adapters like a VME board and add-on accelerator cards for PCs and workstations.

THE VECTOR ARCHITECTURE IS WELL KNOWN in the supercomputer world as a method of achieving high performance for floating-point operations. As open systems become more and more popular in this widely expanding field however, a growing number of users require additional computation power. To meet this trend, designers have significantly improved the processor power of those systems by introducing new architectures. However, applications, such as aerodynamic simulation, structural analysis, and seismic processing, require even higher levels of performance. We consider the μ VP to be an alternative to the high-speed floating-point computation for such open systems as engineering workstations and back-end computing servers. With the μ VP those applications can be executed in a more personal environment.

The design background and implementation of the μ VP required the investment of various hardware facilities and implementation techniques of vector architecture so it could fit on a single CMOS LSI chip. The μ VP has been in production as MB92831-33/45/50 since last November, and volume production parts are available now. 

Acknowledgments

We thank all μ VP project members and parties who were involved with the chip and support environment development. We also thank Richard Cownie of Meiko, who helped us provide performance evaluation information of a real chip.

References

1. J. Dongarra et al., *LINPACK User's Guide*, SIAM, Philadelphia, Penn., 1979.
2. P.M. Kogge, *The Architecture of Pipelined Computers*, Hemisphere Publishing Corporation, 1981.
3. K. Hwang, *Computer Arithmetic Principles, Architecture, and Design*, John Wiley & Sons, New York, 1979.
4. H. Iino et al., "A 289-Mflops Single-Chip Supercomputer," *ISSCC Digest Tech. Papers*, Feb. 1992, pp. 112-113.
5. J.L. Hennessy and D.A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers, Inc., 1990.
6. CS-2 Product Description, Meiko, 1992.



Makoto Awaga, a manager of the μ VP engineering team at Fujitsu's Kawasaki site, has been involved with strategic marketing and product planning of high-end microprocessors and controllers. He has worked on the company's personal computer development and in San Jose, California, on microcomputer marketing and research. Awaga received a BS degree in electrical engineering from Ritsumeikan University, Kyoto.



Hiromasa Takahashi is a researching manager in the Processor Laboratory at Fujitsu Laboratories Limited in Atsugi, Japan. His research interests include microprocessors and architecture for high-speed processing systems. Takahashi received BE and ME degrees in electrical engineering from Chiba University. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan.

Direct any questions concerning this article to Makoto Awaga, Fujitsu Limited, 1015, Kamikodanaka, Nakahara-ku, Kawasaki 211, Japan; or awaga@gmd.ed.fujitsu.co.jp.; or Hiromasa Takahashi, Fujitsu Laboratories Limited, 10-1, Morinosato-Wakamiya, Atsugi 243-01, Japan; or taka@prd.cs.fujitsu.co.jp.

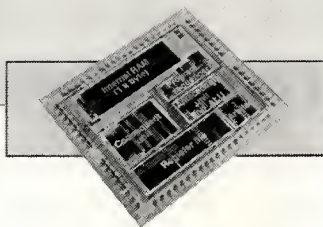
Reader Interest Survey

Indicate your interest in this article by circling the appropriate numbers on the Reader Service Card.

Low 159

Medium 160

High 161



Fuzzy Inference and Fuzzy Inference Processor

Fuzzy inference, a data processing method based on the fuzzy theory, has found wide use, mainly in the control field. Consumer electronics, which accounts for most current applications of this concept, does not require very high speeds. Though software running on a conventional microprocessor can perform these inferences, high-speed control applications require much greater speeds. We have devised a processor that operates at 200,000 fuzzy logic inferences per second. Our design features 12-bit input and 16-bit output resolution.

Kazuo Nakamura

Narumi Sakashita

Yasuhiko Nitta

Ken'ichi Shimomura

Takeshi Tokuda

Mitsubishi Electric
Corporation

As originally proposed by Zadeh,¹ the fuzzy theory treats ambiguous concepts as mathematical numbers or functions. Fuzzy inference, a data processing method based on this theory, has found widespread use, mainly in controllers, and numerous reports attest to its success.² See the Fuzzy theory box on page 39 for further background on this novel and far-reaching concept.

Figure 1 (next page) compares the performance of several implementations of the fuzzy inference. The graph's horizontal axis represents the inference speed measured in units of fuzzy logic inferences per second, or FLIPS. Its vertical axis represents the input resolution. A software approach using a conventional microprocessor can attain up to 1 KFLIPS with 8 to 16 bits of resolution, sufficient for most current consumer electronics applications.

High-speed control applications, however, such as automobile engine control, demand greater inference speed. These applications will require dedicated, large-scale integration that partially or fully performs the inference in hardware. Although several chips have been introduced,³ those operating at higher speeds cannot attain high resolution. Applications demanding sophisticated control will require high resolution as well as high

speed. Here we describe both the fuzzy inference mechanism and the high-speed, high-resolution fuzzy inference processor we have developed to meet these needs.

Fuzzy inference

To understand the fuzzy inference mechanism, we need to take a look at several relevant concepts. We use the example of a representative control system to help us along.

Control system. Figure 2 shows a simple control system. In the figure, "state variable" represents the state of the controlled object. If we take an air conditioner as a controller, for example, the temperature of the room is the state variable. "Reference" is the target value of the state variable—the temperature of a cooler setting. "Disturbance" represents the uncontrollable outside force that changes the state variable—the temperature outdoors.

The control system needs to keep the state variable as closely equal to the reference as possible, or to change the state variable to the reference as quickly as possible. For this purpose, the controller calculates the optimum manipulative variable from the state variable, reference, and disturbance, including their differential and integral values. In the case of the air conditioner, the

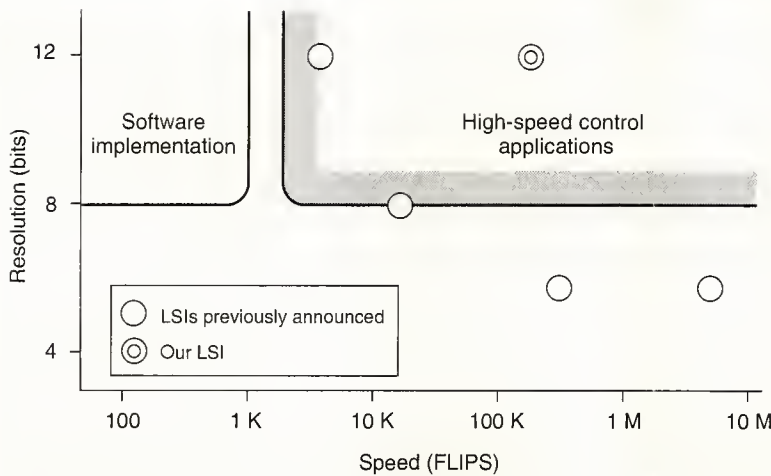


Figure 1. Comparing fuzzy inference performance levels.

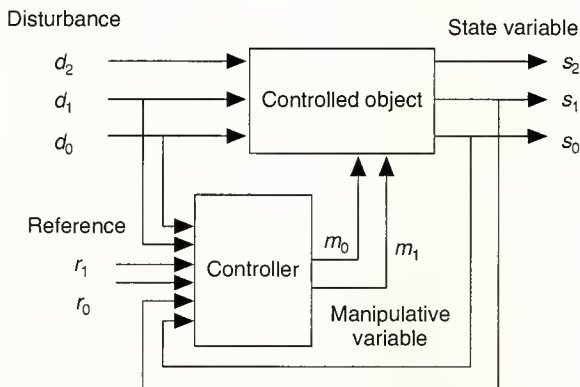


Figure 2. Fuzzy inference control system.

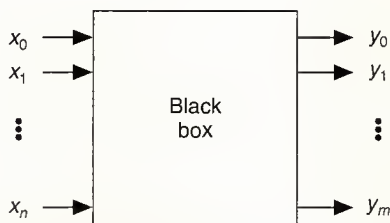


Figure 3. Controller as black box.

manipulative variable would correspond to the motor power.

The key issue for the controller is how to determine the manipulative variable. From this point of view, we can look at the controller as a black box, as shown in Figure 3. At the black box, only the relation between input and output matters. In a conventional controller, a linear or nonlinear function most often defines the relation. Therefore, we calculate output analytically from input according to the function. In a control using fuzzy inference, fuzzy rules define the relation, and output is inferred from input according to the fuzzy inference mechanism.

Fuzzy inference rules. As an example, we can express rules as follows when the number of inputs is n and number of outputs is m :

If $x_0=A_0$ and $x_1=A_1$ and, ..., and $x_n=A_n$ then $y_0=P_0, y_1=P_1, \dots, y_m=P_m$

If $x_0=B_0$ and $x_1=B_1$ and, ..., and $x_n=B_n$ then $y_0=Q_0, y_1=Q_1, \dots, y_m=Q_m$

Here x_0, \dots, x_n are inputs and y_0, \dots, y_m are outputs. $A_0, \dots, A_n, B_0, \dots, B_n, P_0, \dots, P_m$, and Q_0, \dots, Q_m are fuzzy sets. The part of the rule ahead of "then" is called the "antecedent," and the part after it is called the "consequent." We usually define the fuzzy set as a function called the membership function; see Figure 4. In the figure, the horizontal axis represents the input or output. The vertical axis represents the grade, which ranges from 0 to 1 when normalized.

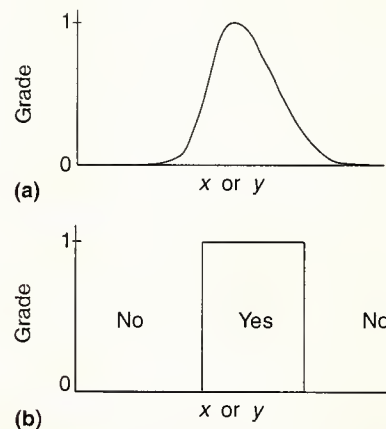


Figure 4. Membership function: fuzzy set (a) and crisp set (b).

Fuzzy theory

Fuzzy theory itself is not "fuzzy" at all, but rather a rigorous mathematical theory. The term *fuzzy theory* is a generic one that encompasses the theories of fuzzy set, fuzzy logic, and fuzzy measure. Fuzzy logic and fuzzy measure theories were derived from fuzzy set theory as established by Lotfi A. Zadeh in 1965.¹

Fuzzy set theory. In classical set theory, an element is either a member of a set or not (crisp set). In fuzzy set theory, elements can be partially a member of a set (fuzzy set).

Let's take temperature as a universal set. In this case, each temperature—20°C or 27°C—is an element. Figure A shows an example of fuzzy and crisp sets for hot temperatures and temperatures higher than 25°C. In both cases, each element has a value called membership or grade that indicates to what degree the element belongs to the set. While the crisp set takes only two values—complete truth (grade = 1, if normalized) and complete falseness (grade = 0)—the fuzzy set takes continuous values from 0 to 1.

Fuzzy set theory is a generalization of classical set theory because it allows both complete truth and falseness. Therefore, any conclusion derived from classical set theory is also valid with fuzzy set theory.

Fuzzy control. A control that uses fuzzy inference as a means to calculate outputs from inputs is called fuzzy control. Fuzzy inference is based on fuzzy theory, which E.H. Mamdani first applied to control fields in 1974.² Control is currently a most active and fruitful field for the application of fuzzy theory, because fuzzy theory is a technically viable and cost-effective discipline. Among the many applications of fuzzy control that have been reported, those for cement kiln control,³ water quality control,⁴ and the automatic train operation system of the Sendai subway⁵ are especially well known.

For a fuller historical perspective of fuzzy control, consult Lee^{6,7} and Brubaker.^{8,9}

1. L.A. Zadeh, "Fuzzy Sets," *Information Control*, Vol. 8, 1965, pp. 338-353.
2. E. H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," *Int'l J. Man-Machine Studies*, Vol. 7, 1975, pp. 1-13.
3. L.P. Holmblad and J.J. Ostergaard, "Control of a Cement Kiln by Fuzzy Logic," in *Fuzzy Information and Decision Processes*,

M.M. Gupta and E. Sanchez, eds., North-Holland, Amsterdam, 1982, pp. 389-399.

4. O. Yagishita, O. Itoh, and M. Sugeno, "Application of Fuzzy Reasoning to Water Purification Process," in *Industrial Applications of Fuzzy Control*, M. Sugeno, ed., North-Holland, Amsterdam, 1985, pp. 19-40.
5. S. Yasunobu and S. Miyamoto, "Automatic Train Operation by Predictive Fuzzy Control," in *Industrial Applications of Fuzzy Control*, M. Sugeno, ed., North-Holland, Amsterdam, 1985, pp. 1-18.
6. C.C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller—Part I," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 20, No. 2, 1990, pp. 404-418.
7. C.C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller—Part II," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 20, No. 2, 1990, pp. 419-435.
8. D. Brubaker, "Fuzzy-Logic Basics: Intuitive Rules Replace Complex Math," *EDN*, June 18, 1992, pp. 111-116.
9. D. Brubaker, "Fuzzy-Logic System Solves Control Problem," *EDN*, June 18, 1992, pp. 121-127.

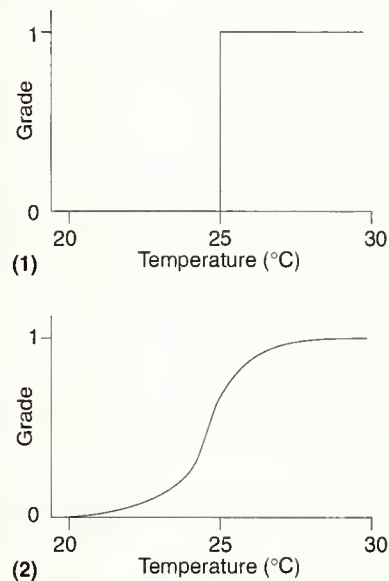


Figure A. Crisp set (1) and fuzzy set (2).

Investigators have proposed several methods for performing fuzzy inference. We will describe the min-max-centroid method, the most popular one. For simplicity's sake, let's consider the case with two inputs, two rules, and one output.

The rules are

- rule 0: If $x_0 = A_0$ and $x_1 = A_1$, then $y = P$
 rule 1: If $x_0 = B_0$ and $x_1 = B_1$, then $y = Q$

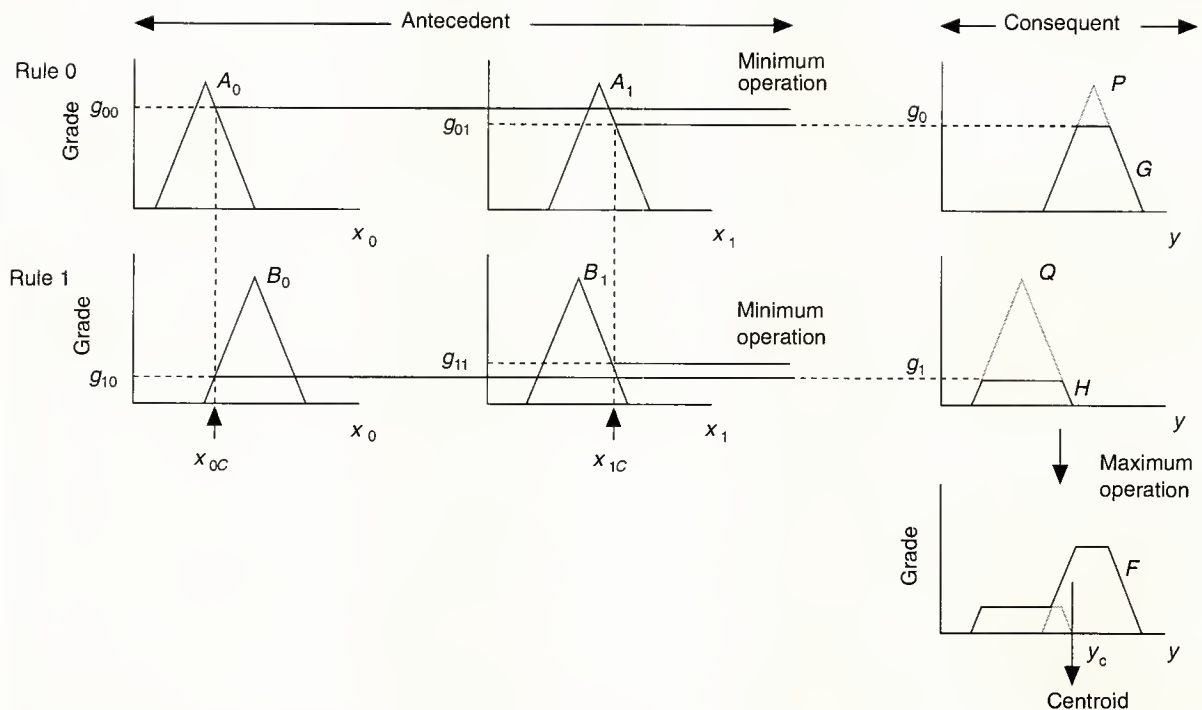


Figure 5. Fuzzy inference schematic.

Figure 5 shows the schematic of the fuzzy inference that results from these rules. As shown, six functions represent membership functions: A_0 , A_1 , P , B_0 , B_1 , and Q . Also, x_{0c} and x_{1c} represent the values of input x_0 and x_1 , respectively, from which the inference is performed. y_c represents the inferred value as output y .

As the first step of the inference, we obtain the value g_{00} , which represents the grade of the equation $x_0 = A_0$, by calculating the value of the function A_0 when $x_0 = x_{0c}$. We obtain the value g_{01} , representing the grade of the equation $x_1 = A_1$, in the same way.

For the semantic *and* operation, we perform the minimum operation between both values by selecting the lesser. As a result, we obtain g_0 , the grade of rule 0 and also g_1 , the grade of rule 1. These comprise the antecedent processing.

Consequent processing involves obtaining the function G by lopping off the part greater than value g_0 from function P . We obtain function H in the same way. To perform the maximum operation, we select the greater values of both functions at each value of y . From this operation, we obtain function F , or the resultant membership function.

Defuzzification, the final processing of the inference, entails calculating the centroid, or center of gravity, of the function F . The result of the defuzzification, y_c , becomes the result of the inference as the value of output y .

The fuzzy concept. Linguistic expressions such as cool, warm, or hot are fuzzy concepts. The fuzzy set is the mathematical expression of the fuzzy concept. The grade indicates to what degree the set includes input or output values. In conventional theory, the degree takes only two values—yes or no. This is the crisp set. In fuzzy theory the set has a continuous value. Figure 4 illustrates this distinction.

However, this recognition is rather complex. If we are considering only fuzzy inference, it may be easier to view the rule as follows:

If x_0 matches A_0 and x_1 matches A_1 and, ..., x_n matches A_n , then output Q as y .

In this case, A_0, \dots, A_n specifies the input range. The grade obtained by substituting current values x_0, \dots, x_n for A_0, \dots, A_n indicates the degree of matching. The antecedent specifies the region of the input n -dimensional space composed of x_0, \dots, x_n . The grade of the rule obtained by the *and* operation indicates the degree that current x_0, \dots, x_n values match the region. The Q specifies the value to be output when x_0, \dots, x_n values match the region. This specifies only a scalar value, though we define it as a fuzzy set. We can regard the calculation of the inference as a way to get a weighted average of values designated in the consequent of each rule, using the grades in the

antecedent of each rule as weighting values.

In any rule, the antecedent specifies a region of the input space, while the consequent specifies the output value when current inputs match the region. The rule set then is a lookup table; fuzzy inference is an interpolation of the table. The main difference from conventional lookup tables is that fuzzy logic defines the input region not crisply but fuzzily and uses a special mechanism called fuzzy inference for the interpolation of the table.

The calculation of a centroid is complex. To simplify the process, we sometimes use singleton functions. With these, consequent membership function does not designate a range but a value. Using these functions, the result of the inference is not so different from cases that use an ordinary function, unless redundancy of the rules exists.

The min-max-centroid method can eliminate the redundancy of the rule set. Consider a rule set having two rules that have the same or nearly the same meaning. Both rules have the same or close consequent membership functions. By calculating the simple weighted average, we duplicate the weighting operation for each rule. Using the min-max-centroid method eliminates this duplication, because the maximum operation selects only one value. At this point, the singleton function is at a disadvantage because it can eliminate redundancy only when consequent membership functions are exactly the same. In the same way, the minimum operation at the antecedent can eliminate the redundancy between the equations (*and* term).

Execution of inference

Next, we want to further investigate the fuzzy inference by exploring antecedent and consequent processing, membership function calculation, and inference reliability.

Antecedent processing. Figure 6 shows an example of a rule set and its resultant membership function. The rule set is composed of rules 0 to 7. The grades of these rules have an assumed value of g_0 to g_7 . As shown, only g_0 , g_2 , g_3 , and g_6 affect the inference. These values are the largest grades among the rules that have the same consequent membership function. This is a characteristic of the min-max-centroid method.

We can calculate grades g_0 , g_2 , g_3 , and g_6 as shown in Figure 7 (next page). Here $\min()$ and $\max()$ represent minimum and maximum operation, while min register and max register represent the accumulator into which the result of the minimum operation and maximum operation are stored.

Consequent processing. Assume that the resultant mem-

Rule 0 :	if $x_0=A_0$ and	$x_1=A_1$ and	$x_2=A_2$	then $y=P_0$
Rule 1 :	if $x_0=B_0$		$x_2=B_2$	then $y=P_0$
Rule 2 :	if	$x_1=C_1$ and	$x_2=C_2$	then $y=P_1$
Rule 3 :	if $x_0=D_0$ and	$x_1=D_1$		then $y=P_2$
Rule 4 :	if $x_0=E_0$	and	$x_2=E_2$	then $y=P_2$
Rule 5 :	if $x_0=F_0$ and	$x_1=F_1$ and	$x_2=F_2$	then $y=P_2$
Rule 6 :	if	$x_1=G_1$ and	$x_2=G_2$	then $y=P_3$
Rule 7 :	if $x_0=H_0$			then $y=P_3$

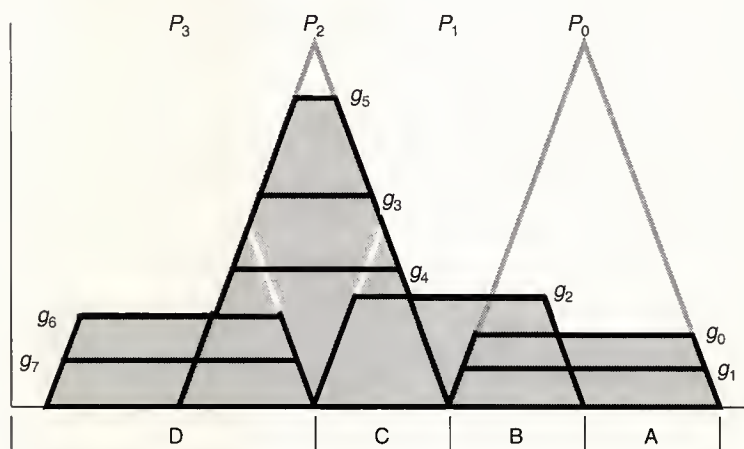


Figure 6. Sample rule set and resultant membership function.

bership function $f(y)$ is defined by the discrete values of $f(0), f(1), f(2), \dots, f(n-1), f(n)$. In this case, the centroid of $f(y)$ is defined with the quotient of the division P by Q , where P and Q are defined as follows:

$$P = f(0) * 0 + f(1) * 1 + \dots + f(n-1) * (n-1) + f(n) * n$$

$$Q = f(0) + f(1) + f(2) + \dots + f(n-1) + f(n)$$

Here, we can express P as follows:

$$P = \{f(n)\} \\ + \{f(n) + f(n-1)\} \\ + \dots \\ + \{f(n) + f(n-1) + \dots + f(2)\} \\ + \{f(n) + f(n-1) + \dots + f(2) + f(1)\}$$

We can define P_j and Q_j as follows:

$$P_j = \{f(n)\} \\ + \{f(n) + f(n-1)\} \\ + \dots$$

```

(initialization)
0  0 -> max reg; 255 -> min reg
(rule 0)
1  min(A0(x0), min reg) -> min reg
2  min(A1(x1), min reg) -> min reg
3  min(A2(x2), min reg) -> min reg
4  max(min(A2(x2), min reg), max reg) -> max reg
5  255 -> min reg
(rule 1)
6  min(B0(x0), min reg) -> min reg
7  min(B2(x2), min reg) -> min reg
8  max(min(B2(x2), min reg), max reg) -> max reg
9  255 -> min reg; 0 -> max reg
(rule 2)
10 min(C1(x1), min reg) -> min reg
11 min(C2(x2), min reg) -> min reg
12 max(min(C2(x2), min reg), max reg) -> max reg
13 255 -> min reg; 0 -> max reg
⋮
⋮
⋮
(rule 6)
27 min(G1(x1), min reg) -> min reg
28 min(G2(x2), min reg) -> min reg
29 max(min(G2(x2), min reg), max reg) -> max reg
30 255 -> min reg
(rule 7)
31 min(H0(x0), min reg) -> max reg
32 max(min(H0(x0), min reg), max reg) -> max reg
33 255 -> min reg; 0 -> max reg

```

Figure 7. Grade calculation schematic

$$\begin{aligned}
 &+ \{f(n) + f(n-1) + \dots + f(j+1)\} \\
 &+ \{f(n) + f(n-1) + \dots + f(j+1) + f(j)\} \\
 Q_j &= f(n) + f(n-1) + f(n-2) + \dots + f(j)
 \end{aligned}$$

We can then express P as follows:

$$P = Q_n + Q_{n-1} + Q_{n-2} + \dots + Q_2 + Q_1$$

Since Q_j is the temporary result in calculating Q , the following steps give us P and Q :

- Set $j = n$, $Q_j = 0$, and $P_j = 0$.
- Add Q_j to P_j and $f(j)$ to Q_j .
- Subtract 1 from j .
- Repeat steps 2 and 3 until j becomes negative.
- Q is obtained as Q_j and P is obtained as P_j .

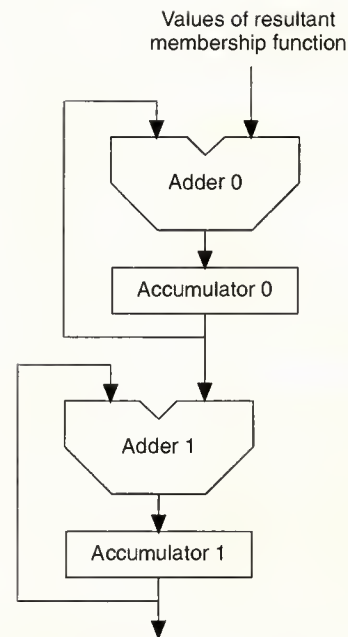


Figure 8. Two-stage adder/accumulator.

A two-stage addition/accumulation can easily perform these steps either through software or by using the hardware shown in Figure 8.

In the example shown in Figure 6, we can obtain P and Q by using the algorithm composed of the following nine processes. This is the algorithm we employed in designing our fuzzy inference processor. It lets us pipeline both the antecedent processing (rule calculation and minimum/maximum operation) and the consequent processing (accumulation).

1. Arrange the rules in the descendent order of the values of their consequent membership functions. (In Figure 6, rules 0 to 7 are already arranged according to the larger functions P_0 to P_3).
2. Calculate grade g_0 and g_1 and perform the maximum operation between these values.
3. Perform a two-stage accumulation in area A of the function $f_0(y)$ expressed as follows:

$$f_0(y) = \min\{P_0(y), g_0\}.$$

4. Calculate grade g_2 .
5. Perform a two-stage accumulation in area B of the function $f_1(y)$ expressed as follows:

$$f_1(y) = \max[\min\{P_0(y), g_0\}, \min\{P_1(y), g_2\}].$$

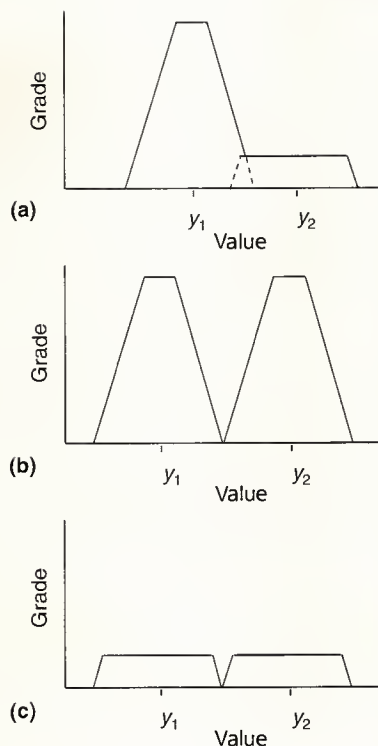


Figure 9. Resultant membership function: no conflict (a); conflict, with large variance (b); conflict, no rule has large grade (c).

6. Calculate grades g_1 , g_2 , and g_3 and perform the maximum operation among these values.
7. Perform a two-stage accumulation in the area C of the function $f_2(y)$ expressed as follows:

$$f_2(y) = \max[\min[P_1(y), g_2], \min[P_2(y), g_3]].$$

8. Calculate grades g_6 and g_7 and perform the maximum operation between these values.
9. Perform a two-stage accumulation in the area D of the function $f_3(y)$ expressed as follows:

$$f_3(y) = \max[\min[P_2(y), g_3], \min[P_3(y), g_6]].$$

Membership function calculation. For a fast inference it is also important to calculate the value of membership function. Using a lookup table may be best when software performs the inference, even though it requires a large memory. The literature cites an example of the hardware implementation.⁴

Inference reliability. Though the conventional fuzzy inference mechanism has no facility to evaluate the reliability

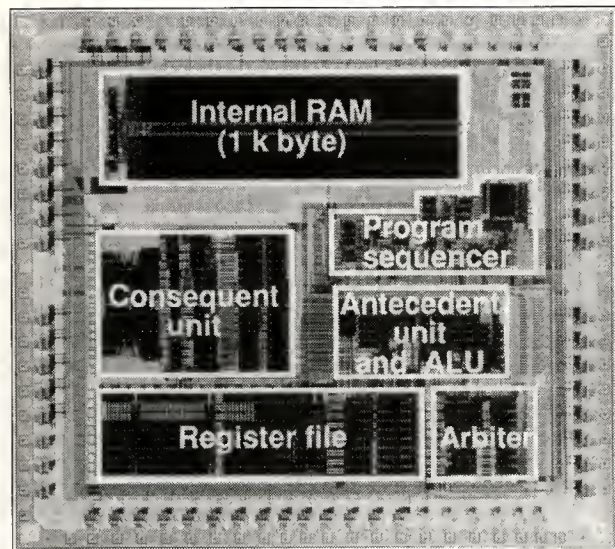


Figure 10. Photomicrograph of fuzzy inference processor.

of the inference, practical applications need such a reliability evaluation. The following two mechanisms serve this purpose.

Calculation of variance. The resultant membership functions shown in Figures 9a and 9b have the same centroid value. In both functions, the rule causing the left trapezoid asserts to output the value y_1 , while the rule causing the right trapezoid asserts to output the value y_2 . At the function in Figure 9a, no conflict exists because the grade or "closeness to truth" of the former rule is large but the grade of the latter is small. On the other hand, at the functions in Figure 9b, both grades are great, so both rules conflict with each other. Checking the variance can distinguish these cases. A three-stage adder/accumulator unit can establish the variance.⁴

Calculation of maximum grade. Comparing the two resultant membership functions shown in Figure 9a and 9c, we see that the maximum value of the former is relatively great, but the value of the latter is very small. The result from the former thus has considerable "closeness to truth," but the result of the latter has little. We can easily check this by estimating the maximum grade of the rule.

Fuzzy inference processor

Now that we understand some of the concepts underlying it, we can move on to describe the actual fuzzy inference processor we have created.

Performance. Figure 10 shows a photograph of the fuzzy inference processor and Table 1 (next page) gives its specifications.⁵ The processor can perform an inference at 200 KFLIPS with 12-bit input resolution and 16-bit output resolution. For these calculations, we use a typical inference having about

Table 1. Performance of fuzzy inference processor.

Parameter	Specification
Inference speed	200 KFLIPS (20 rules with 2 inputs/1 output)
Input resolution	12 bits
Output resolution	16 bits
Membership function shape variations	About 30,000
Number of rules	More than 15,000 (2 inputs/1 output)
Clock rate	20 MHz
Power supply	+5V
Package	80-pin package

20 rules with two to four inputs and a single output. We determined these performance levels by researching the engine control as follows.

For automobile engine control, inhaled air mass is the most significant input. Currently, designers sample this input at 8 bits; they will sample it at 10 to 12 bits in the near future. An eight-cylinder engine running at 10,000 rpm is ignited 1,333 times per second. In determining ignition delay timing, we find that performing several inferences in the interval between ignitions will require an inference speed of greater than 10 KFLIPS. That makes 100 KFLIPS a reasonable target for the inference in light of other outputs such as fuel mass.

Inference processor hardware. See Figure 11 for a block diagram of this processor. The antecedent and consequent units are the special hardware added for the fuzzy inference. The arithmetic logic unit, register file, internal RAM, and program sequencer constitute a conventional microprocessor. The processor has two processor modes: master (for stand-alone use) and slave (for coprocessor use). Using the arbiter, the master processor can access the register files, internal RAM, and external memory without resorting to off-chip hardware.

The data width for processing in the ALU, internal RAM, and register file is 16 bits, a rate that was determined to preserve conformity with the master processor and efficiency of the instruction code. Antecedent processing, performed in the antecedent unit, varies depending on the number of rules

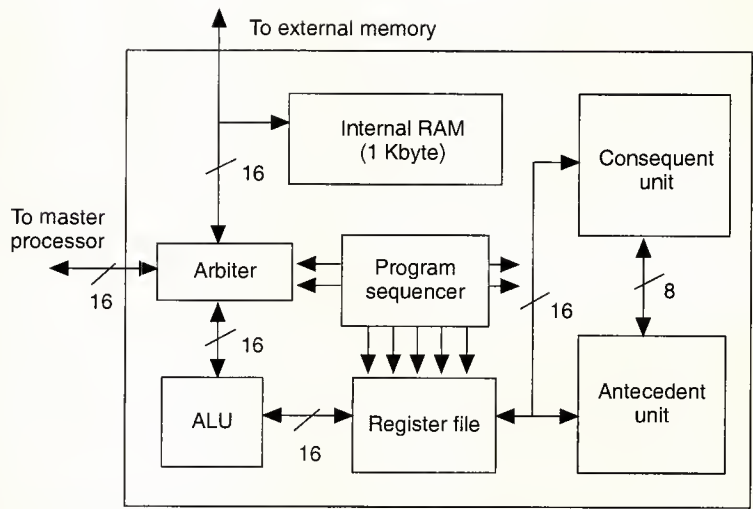
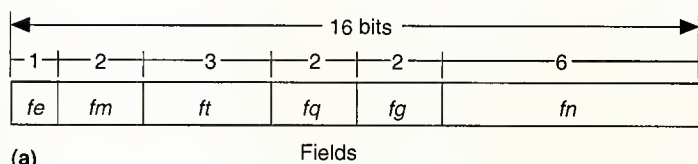
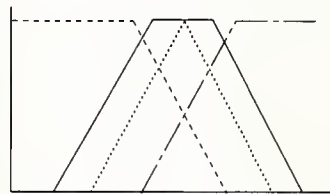


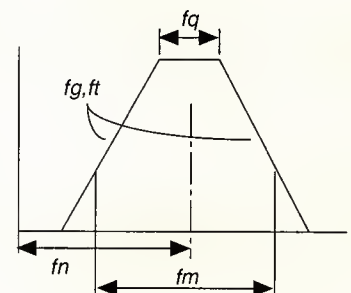
Figure 11. Fuzzy inference processor block diagram.



(a) Fields



(b)



(c)

Figure 12. Membership function and function parameter: 16-bit function parameter (a), function shapes (b), and fields (c).

and inputs. Accumulation for the centroid calculation, performed in the consequent unit, is usually realized by a fixed sequence. According to our design, the program composed of the rule instructions controls the antecedent unit, while the hardwired logic controls the consequent unit.

Membership function generator. The membership function generator calculates the value of the membership function. As shown in Figure 12b, this dedicated hardware generates four kinds of function shapes based on a trapezoid. The 16-bit data called the function parameter specifies the function (see

Figure 12a). The parameter consists of six fields: fn for specifying the center position of the function, f_q for the length of the top side if the function is trapezoid, fm for the scale factor of the horizontal direction, fg for specifying the inclination, and ft for both specifying the function-shape and modifying the inclination of the oblique sides if the function shape is trapezoidal or triangular. Using the function parameter, more than 30,000 variations of membership function can be generated. The optional fe field allows for changing the shape of the oblique sides (for example, from straight line to S-shaped) by referring to a lookup table. A microprogram included in the program sequencer controls this change.

Antecedent unit and rule instruction. The antecedent unit performs the grade calculation of the rule when the program designates a rule instruction. Included in the register file are 16 data registers that hold the input values of the inference which were stored there in advance. These instructions fall into two categories: crule (pronounced see-rule), which triggers the accumulation for the centroid calculation, and rule, which does not. Figure 13 compares the formats of these instructions.

Each instruction consists of an opcode and a register list followed by the membership function parameters. The register list represents the input used for the inference. Paired with each data register is each corresponding bit in the register list; setting the corresponding bit to 1 designates an arbitrary data register. The membership function parameters individually specify the shape of the membership functions. The number of the parameters therefore corresponds to the number of bits that are each set to 1 in the register list of the rule instructions.

Crule instructions have one additional function parameter: The last parameter designates the consequent membership function. In the crule instruction, the opcode includes the accumulation constant—represented as cc in Figure 13—that specifies the repeat count of accumulation. The centroid calculations use these parameters and constant values.³

Figure 14 shows the block diagram of the antecedent unit. The membership function generator generates the values of the membership function designated by the input value and function parameter. The min operator performs the minimum operation between the function value and the value in the min register, which is initially set to the maximum value of the grade at the end of the previous rule or crule instruction execution. The min register stores the result.

The max operator performs the maximum operation between the result of the minimum operation and the value in the max register, which is cleared to 0 at the end of the previous crule instruction execution. The max register stores the result. After performing the maximum operation, the antecedent unit sets the min register to the maximum value of the grade. When executing a rule instruction, the unit performs no additional operations, and the result of the execution remains in the max register.

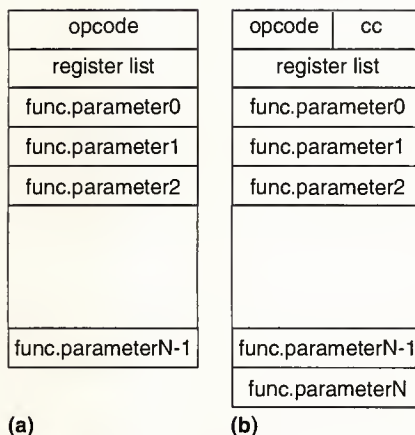


Figure 13. Rule instruction: rule (a) and crule (b).

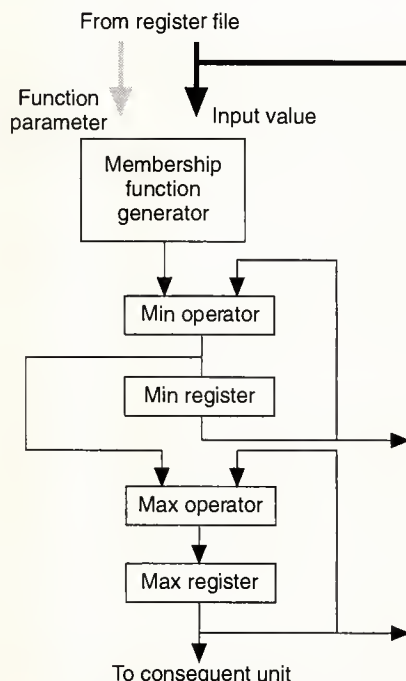


Figure 14. Antecedent unit block diagram.

When executing a crule instruction, the antecedent unit performs an additional operation. It transfers the last function parameter to the consequent unit along with the value in the max register and the accumulation constant in the instruction register. After the transfer, the unit clears the max register to 0.

Consequent unit. This unit is divided into the min/max

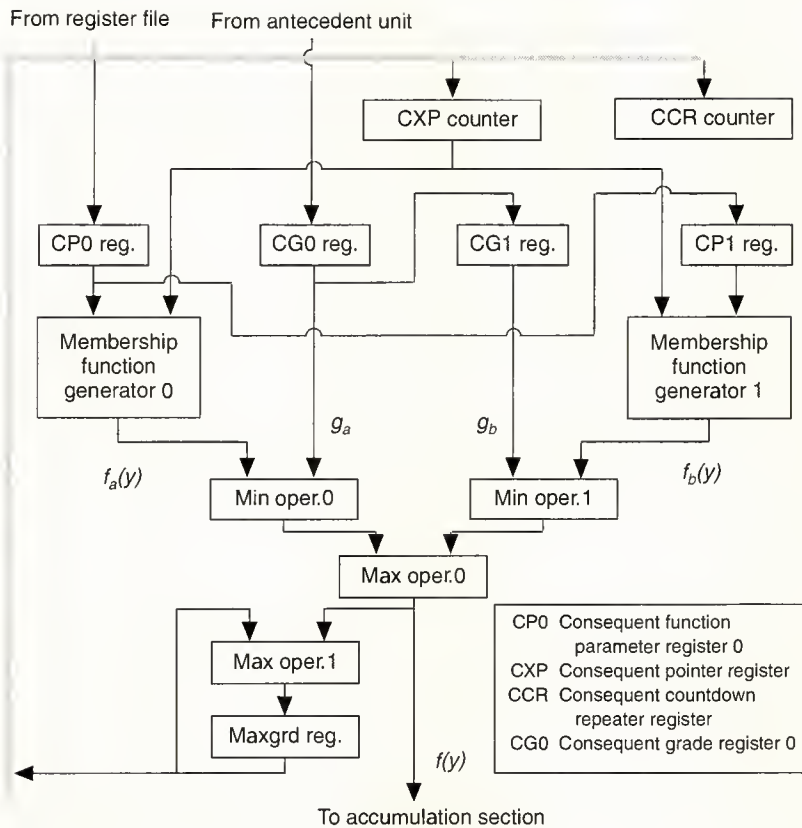


Figure 15. Min-max section block diagram.

1	inicp	MaxY
2	rule	[r0,r1,r2],A ₀ ,A ₁ ,A ₂
3	crule	[r0,r2],B ₀ ,B ₂ ,P ₀ ,CnstA
4	crule	[r1,r2],C ₁ ,C ₂ ,P ₁ ,CnstB
5	rule	[r0,r1],D ₀ ,D ₁
6	rule	[r0,r2],E ₀ ,E ₂
7	crule	[r0,r1,r2],F ₀ ,F ₁ ,F ₂ ,P ₂ ,CnstC
8	rule	[r1,r2],G ₁ ,G ₂
9	crule	[r0],H ₀ ,P ₃ ,CnstD
10	cntrd	OfstY
11	vrns	ThvV
12	cmpgrd	ThvG

Figure 16. Sample inference program.

section and the accumulation section. Figure 15 (next page) shows the min/max section. The following equation expresses $f(y)$, which is the output of the max operator 0:

$$f(y) = \max[\min[f_a(y), g_a], \min[f_b(y), g_b]]$$

where $f_a(y)$ and $f_b(y)$ are the membership functions designated by the parameters in the CP0 and CP1 registers, g_a and g_b are the grades stored in the CG0 and CG1 registers, and y represents the value of the CXP counter.

The max operator 1 performs the maximum operation between the value in the maxgrd register and the value of $f(y)$. The consequent unit clears the maxgrd register to 0 in advance of the inference, and stores it with the value selected by the max operator 1, during the calculation of $f(y)$.

The accumulation section contains the three-stage adder/accumulator unit. It accumulates the value of the resultant membership function calculated in the min/max section. See Nakamura et al.⁴ for details.

Inference program. Providing the rule instruction makes this program quite simple. Figure 16 shows an inference of the rule set given in Figure 6. This example assumes that inputs of x_0, \dots, x_2 are already stored into data registers of r_0, \dots, r_2 .

In the figure, the INICP instruction initializes the registers. The max register, CG0 register, and three accumulators included in the three-stage adder/accumulator unit are initialized to 0. MaxY, which

is stored into the CXP counter, represent the maximum value of y . Rule instructions and crule instructions calculate the grade of the rules and the resultant membership function according to the algorithm described earlier. In these instructions, registers in brackets represent the register list. A_0, \dots, H_0 represent the function parameters of antecedent membership functions. P_0, \dots, P_3 represent the function parameters of consequent membership functions. CnstA, ..., CnstD represent the accumulation constants. These values are the widths of the area $A, \dots, \text{area } D$ in Figure 6.

Each rule/crule instruction calculates a grade of the rule and performs the maximum operation between the grade and max register. Only crule instructions trigger accumulation, which they do by transferring the accumulation constant, the value of the max register, and the consequent membership function to the consequent unit. After the transfer, the processor clears the max register to 0.

As shown in Figure 17, the accumulation is pipelined with the execution of the rule instructions.⁴ The cntrd instruction calculates the centroid, then adds the value of OfstY to it. OfstY is the offset value that is added to the centroid. The

vms instruction calculates and checks the variance. ThvV represents the threshold value of the variance. The cmpgrd instruction checks the maximum grade by comparing the maxgrd register with the value of ThvG.

Execution time. The execution time of the rule instruction specifying N data registers is $N+1$ clock cycles. For the crule instruction, it is $N+2$ clock cycles. The total execution time of inicp, cntrl, vms, and cmpgrd is 44 clock cycles. The accumulation time taken by the three-stage adder/accumulator unit is 64 clock cycles when the maximum value of y is 64. The inference time shown in Figure 16 becomes 73 clock cycles if accumulation is performed in the execution time of the rule instructions. However, in this case, accumulation time may be greater than the execution time of the rule instructions. So, inference time becomes about 100 clock cycles when the maximum value of y is 64.

In the same way, with up to about 20 rules having two to four inputs, the inference time depends on the accumulation time and becomes about 100 clock cycles. This corresponds to 200 KFLIPS performance with a clock of 20 MHz. When the processor performs inferences with more complex rule sets, the inference time depends on the execution time of the rule instruction.

WE CAN LOOK UPON THE FUZZY inference data processing method as an interpolation of a lookup table. It is also a method to define and calculate a nonlinear function. Most significantly, its data processing mechanism can be described by the linguistic expression used in normal speech. The support tools that help to define rule sets or membership functions are important for taking advantage of fuzzy inference. As dedicated LSIs and support tools become available, we believe fuzzy inference will find wider application as it substitutes for conventional data processing. ■

Grade calculated in antecedent unit		--	g_6	g_5	g_2	g_0
Contents of	CG0	g_6	g_5	g_2	g_0	0
	CG1	g_5	g_2	g_0	0	--
Specified functions by	CP0	$P_3(y)$	$P_2(y)$	$P_1(y)$	$P_0(y)$	--
	CP1	$P_2(y)$	$P_1(y)$	$P_0(y)$	--	--

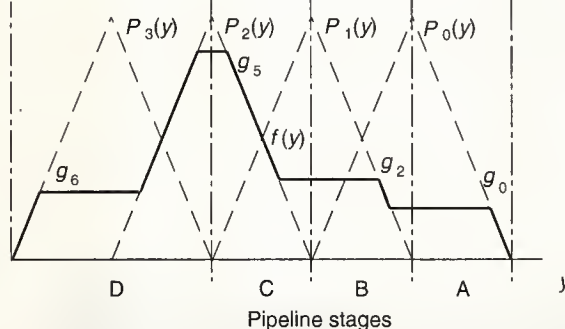


Figure 17. Consequent membership functions and pipelined operation.

Acknowledgments

We gratefully acknowledge Hiroyoshi Komiya, Akira Iwase, and Osamu Tomisawa for their encouragement and advice.

References

1. L.A. Zadeh, "Fuzzy Set," *Information and Control*, Vol. 8, 1965, pp. 338-353.
2. E. H. Mamdani and S. Assilian, "An Experiment in Linguistic Synthesis with a Fuzzy Logic Controller," *Int'l J. Man-Machine Studies*, Vol. 7, 1975, pp. 1-13.
3. H. Watanabe, W. D. Dettloff, and K. E. Yount, "A VLSI Fuzzy Logic Controller with Reconfigurable, Cascadable Architecture," *IEEE J. Solid-State Circuits*, Vol. 25, No. 2, 1990, pp. 376-382.
4. K. Nakamura et al., "A 12-Bit Resolution 200 KFLIPS Fuzzy Inference Processor," *IEICE Trans. Electronics*, Vol. E76-C, No. 5, July 1993, pp. 1102-1111.
5. K. Nakamura et al., "A 12-Bit Resolution 200 KFLIPS Fuzzy Inference Processor," *ISSCC Digest of Technical Papers*, Feb. 1993, pp. 182-183.



Kazuo Nakamura is a senior engineer at Mitsubishi Electric's Kita-Itami works. During this project, he worked at its LSI laboratory, where he was engaged in application engineering of LSI and design engineering of general-purpose processors, fuzzy inference processors, and their support software.

Nakamura received BS and MS degrees in electrical engineering from the Tokyo Institute of Technology.



Narumi Sakashita is a research engineer at Mitsubishi's Semiconductor Research Laboratory. His interests include design engineering of digital signal processors and fuzzy inference processors.

Sakashita received a BS in electrical engineering from Kansai University, Osaka, Japan. He is a member of the Institute of Electronics, Information, and Communication Engineers.



Yasuhiko Nitta is a research engineer at Mitsubishi's Semiconductor Research Laboratory where he has been engaged in the design engineering of general-purpose processors and fuzzy inference processors.

Nitta received a BS in electrical engineering from Waseda University, Tokyo, Japan.



Ken'ichi Shimomura is a research engineer at Mitsubishi's Semiconductor Research Laboratory where he has been engaged in the design engineering of general-purpose processors and fuzzy inference processors.

Shimomura received BS and MS degrees in physics from Osaka University.

IEEE COMPUTER SOCIETY PRESS

DIGITAL IMAGE PROCESSING, 2nd Edition

edited by Rama Chellappa

This collection of papers examining digital image processing is a revised and updated version of the first edition, published in 1985. It concentrates on discussions of image compression, explorations into image enhancement with an emphasis on median and related non-linear filtering of images, and investigations of image restoration to preserve discontinuities while smoothing or restoring.

The tutorial includes an overview of all five sections and a total of 47 papers of which 35 are new to this edition. The text explores models for:

- | | |
|----------------------------------|-------------------------------|
| * Image Data | * Filters in Image Processing |
| * Adaptive Restoration of Images | * Image Coding Techniques |
| * Vector Quantization | * Image Sequence Compression |

Sections: Image Models, Image Enhancement, Image Restoration, Image Data Compression, Emerging Topics, Bibliography.

816 PAGES. APRIL 1992. HARDCOVER. ISBN 0-8186-2362-4.
CATALOG # 2362-01 — \$80.00 MEMBERS \$64.00



**To order call: 1-800-CS-BOOKS
or FAX 714-821-4010**



Takeshi Tokuda is manager of the R&D planning section of Mitsubishi's System LSI Laboratory. His interests include development of CMOS logic LSI and digital signal processor.

Tokuda received BS and MS degrees in communications engineering from Osaka University. He was also a visiting scholar at Stanford University, where he worked on VLSI processor design. He is a member of the Institute of Electronics, Information, and Communication Engineers.

Direct questions concerning this article to Ken'ichi Shimomura, Mitsubishi Electric Corporation, Semiconductor Research Laboratory, 4-1 Mizuhara, Itami, Hyogo 664, Japan; or at simomura@micro.lsi.melco.co.jp.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 156

Medium 157

High 158



Special Report: HDTV Research in Japan

High-definition television has generated a great deal of interest in a variety of fields in recent years. While research in the United States and Europe has focused on a digital approach, Japan has already developed an analog system. It is now applying its knowledge toward development of digital HDTV.

David K. Kahaner

US Office of Naval
Research, Far East

kahaner@cs.titech.ac.jp

[David Kahaner is on assignment with the US Office of Naval Research. He generally comments on activities in East Asia for inclusion in the Software Report column. Since we felt readers would be interested in a detailed description of Japan's research on high definition television, we also offer this special report. His comments are his own; they do not express any official policy.—Ed.]

Most research on high-definition television in the West has focused on digital, rather than analog, technology. As the digital method is more flexible, many in the field believe that these efforts will ultimately succeed in making digital HDTV products more successful and profitable than those using analog technology. While digital HDTV is not currently available, Japan has developed an HDTV system based on analog technology (known as Hivision) that has been in use since 1991. As there are many technical problems shared by both types of systems, research in digital HDTV has also been active in Japan. The Japanese have estimated that digital receivers will not become fully integrated until the next century. Until then, Japanese manufacturers have their potentially large market to themselves.

I recently participated in NHK's annual open house. Thousands of visitors filed through displays highlighting interesting new research and

prototype activities, many of which centered on technology aimed at HDTV. Although the crowds made serious discussion about any of the projects difficult, abstracts for each display were available. A look at some of the more interesting displays provides an overview of Japanese research in this area.

Broadcast, transmission, and reception

Research to develop high-quality systems for broadcast, transmission, and reception forms the core of NHK's work on Hivision. I have chosen several projects in each of these areas to highlight.

Attaining high picture quality through MUSE. MUSE is a transmission system that enables broadcasting on a single channel, the same as with current satellite broadcasts, using bandwidth compression technology. Experimental Hivision broadcasts using the MUSE method now take place eight hours a day. Hivision video signals equivalent to five times the current TV signals provide Hivision video to viewers throughout the country.

Integrated service digital broadcasting. Analog waves currently carry TV and audio signals, with digital waves likely in the future. Compression technology makes multichannel TV with a single broadcasting wave possible. ISDB is a broadcast system capable of transmitting various digitized services by integrating multilayers on a single radio wave.

**Development of a system to
allow viewing of 3D images
without special glasses is a basic
research theme.**

Joining Hivision with computers will allow additional combinations of multifunctional media and information services.

Digital broadcasting on land. Weakening of reception signals by multipath phased propagation represents a significant obstacle to receiving land broadcasting waves on mobile bodies, such as automobiles. Conventional methods of transmission make complete duplication of signals by the receivers difficult. Therefore, a new method of transmission based on digital audio broadcasting is under consideration. This approach involves multiplexing multiple stereo programs within a wide band modulated through orthogonal frequency division multiplexing. In addition, this method also allows for effective mobile transmissions.

42-GHz-band Hivision digital FPU. An analog type Hivision repeater transmission unit using a 42-GHz-band radio wave is already in use. However, rainfall greatly weakens the 42-GHz band, limiting transmissions using this method to relatively close ranges. Accordingly, researchers developed the 42-GHz-band Hivision FPU to expand the range of high picture quality through digital transmissions.

Demand-access optical CATV system. In the near future, multichannelization of satellite broadcasts will result in high growth of new broadcast services, led by Hivision. Researchers are investigating fiber-to-the-home broadcasts that would prove suitable for satellite broadcasting.

Development of a high-quality, inexpensive multichannel video distribution service, which makes the FM-FDM type of demand-access optical CATV system possible, is complete.

Stratified digital transmission. In broadcasts using the digital transmission method, reception often degrades suddenly due to obstacles and strong rainfall. When applying digital transmissions to broadcasting, it is preferable for the degradation in reception quality to occur slowly. The stratified transmission method attempts to prevent sudden degradation.

Hivision reception quality measuring devices. Investigators have developed devices that measure high frequency and MUSE signal characteristics to evaluate Hivision reception

quality.

Mobile receiving system carried aboard aircraft. Since satellite broadcasting covers wide areas, reception in mobile bodies provides an ideal application. Development of a mobile receiving system for ships, trains, sightseeing buses, and cars is complete. The current project involves development of a system for aircraft to allow maximum use of the wide-area feature of satellite broadcasting.

Portable satellite news-gathering system using flat-surface antennae. Developments in digital technology have led to decreased size for high-performance SNG systems. As a result, researchers expect phenomenal improvements in system operability and further perfection of broadcasting program material. Investigators are proceeding with the development of a portable SNG employing flat-surface antennae and solid-power amplifiers, allowing convenient transportation of the system.

Mobile receivers for FM multiplex broadcasting. Digital signals are multiplexed via pauses in the voice signals of existing stereo broadcasting. Efficient use of frequencies allows assembly of an inexpensive system. Broadcasting via FM multiplex mobile receivers permits travelling vehicles to receive a variety of information (such as traffic conditions, news, weather, and stock prices) in real time.

Three-dimensional technology

Television in three dimensions has held the imagination of the entertainment industry for a long time. NHK hopes to couple 3D television with Hivision, and is developing several products with this goal in mind.

3D Hivision without glasses. Development of a system to allow viewing of 3D images without special glasses is a basic research theme in the area of 3D television. In line with this goal, researchers have developed the 70-type liquid crystal projection 3D Hivision display that improves the quality of 3D images as well as a simulator that electrically reproduces multieye 3D images, using a biconvex lens.

Small Hivision camera for 3D photography. A small camera with the properties of easy operation and mobility would allow the use of 3D television in a wide variety of fields (medicine, for example). In addition, a camera with a short distance between the camera lenses (approximately the distance between human eyes) would produce more natural 3D images. Investigators have developed such a Hivision camera and mount.

Audio image distance control for 3D television. The stereo-photographic impact of 3D television would increase dramatically if the audio images also jumped out three dimensionally. Mixing the high points of sound pressure provides such a sense of distance of audio images (achieved by localizing the sound points in a single point within a vacuum). The recently developed audio image control system has both audio image dimensionality and continuous position shifting.

HARP technology

NHK hopes that use of HARP technology will lead to increased sensitivity in cameras used to shoot Hivision programs.

Hivision super-HARP handy cameras. Improvement in the characteristics of the HARP camera tube would aid development of a supersensitive camera capable of coping with the diversification in Hivision program production. The supersensitive, high-quality camera with a 1-inch, 55-type super-HARP tube (static accumulation/electrostatic deflection) developed last year has already photographed the Aurora.

Current experiments with the new small, 2/3-inch, 55-type super-HARP tube have led to development of a supersensitive Hivision handy camera.

High-sensitivity imaging through intermittent scanning. The shutter speed of an ordinary TV camera averages 1/60 second. By slowing down shutter speed and scanning beams intermittently over a set time period, even dark objects photograph easily. Pairing this approach with the cascading effect (the essence of the HARP tube) produces extremely high sensitivity. By connecting this intermittent scanning adapter to the HARP camera (525-line type), the possibility of photographing very slow moving objects exists.

Camera technology

While transmission and reception of programs is important to the quality of Hivision, even the highest quality equipment is only as good as the image provided. Superior images require superior cameras. NHK's research on camera technology applies to Hivision as well as more general areas of photography.

High-performance Hivision 4-panel-type, charge-coupled device image experiment. To provide more Hivision programs, industry requires a small camera with excellent picture quality and high mobility. To satisfy this demand, researchers must study high-performance image elements along with imaging methods suitable for high resolution. Previous cameras used the tricolor resolution prism. However, by combining a color resolution prism split in two with four 2/3-inch, 1.3-million-dot CCDs, investigators produced a 4-panel-type (quad-CCD) image-testing system. This approach produced superior resolution, sensitivity, and dynamic range.

Intelligent robot cameras. Producing programs effectively often relies on the use of robot cameras, both for broadcasting programs and in news studios. However, the conventional robot cameras cannot follow the movements of objects being photographed. For example, when an object shifts position and moves out of the picture angle, robot cameras cannot correct the picture angle.

However, the intelligent robot camera automatically corrects the picture angle (camera frame). Using image processing technology, smooth camera following and real-time corrections are possible.

***Conventional robot cameras
cannot follow the movement of
objects. The intelligent robot
camera automatically corrects
the picture angle.***

Automatic animal-tracking camera. When shooting the ecology of wild animals, the presence of a photographer becomes a big obstacle for the animal with a strong sense of caution. Problems in obtaining effective images of the animal's natural behavior result. Automatic photography with unmanned cameras sometimes yields good images. However, wild animals often refuse to cooperate with stationary cameras and move away from the field of vision.

Development of the automatic tracking camera aimed to remedy this difficulty. This device captures the animal as it appears. The camera then tracks it, recording its actions as it moves in a wide area, giving us a more realistic record of its behavior. This new technology provides a new realm in animal photography. Furthermore, it would provide an effective surveillance tool if used in crime prevention systems.

Electronically driven, super-high sensitivity camera elements. Past research on various types of super-high sensitivity camera devices for photography at ultra-low lighting levels concentrated on those joining the image intensifier and the CCD, and those using fiber optics. Waves in the fiber plates and the granular condition of the fluorescent surface degrade in level of resolution levels and picture quality. Current research centers on a new, electronically driven device that does not use fiber plates or fluorescent surfaces.

Image composition and picture quality

Current program composition packages do not always meet the quality standards required by Hivision. As a result, a variety of techniques to boost composition quality are under development.

Hivision domain-extracting system. Production of TV programs and movies depends heavily on image composing technology, particularly domain extraction. Until now the Chromaki method has formed the basis of this technology. However, this method requires a special background, known as blue back, which precludes application to natural images. An extracting technology for natural imaging would prove indispensable in program production.

Researchers previously developed a procedure to cut an

***A continuous voice recognition
system would automate
superimposition of voices
enunciated by an indefinite
number of speakers in broadcast
programs.***

object from a randomly moving image, automatically describing the object in a general way. This procedure forms the basis of the new extracting system for use with Hivision. It allows the composite processing of program production, and can also construct an image parts database for time-space editing of images in an image production environment (desktop program production).

Standard animated image for evaluation of Hivision systems. In the overall evaluation of picture quality of Hivision systems and equipment, it is essential to use a variety of picture patterns. Different evaluation objectives require different pictures. Furthermore, uniform evaluation results require use of a common picture. Therefore, the industry requires picture standards.

In response to this need, the Broadcast Technology Association is selecting standard pictures. Once compiled and distributed, these picture sets will become the standard Hivision animated pictures used for system evaluation.

Video computer desktop program production. DTPP will improve programming functions, enlarge video displays, improve program quality, and conserve energy. It aims to simplify the process of producing programs and provides features such as editing, fabricating, and mixing. Simplified access to video materials and information such as camera parameters and lighting conditions, all while remaining at one workstation, would result.

Movement compensating sequential scanning conversion. Current TV interlaced scanning methods transmit every other scanning line. Interlaced scanning compresses the bandwidth of TV signals and is useful in effectively broadcasting radio waves. However, because the thin horizontal lines and oblique lines flicker, they interfere with picture quality.

To correct this interference, Clearvision stores received video pictures in memory, intermixes current images with those preceding them, and then displays the images without gaps between scanning lines (sequential scanning conversion display). This process, however, results in only small image

quality improvements. A variation mixes a current image with the image just corrected, resulting in higher quality images with less interference to picture quality.

Voice

Television presents many opportunities for the application of voice recognition. As a result, this topic receives a great deal of attention from numerous research groups, including NHK.

Real-time voice speed converter receiving system. Quality voice broadcasting requires a voice speed converter system that automatically converts fast speech to slower speech. The system divides voice inputs into a soundless sector (a pause), voiceless sectors (consonants), and voice sectors (vowels). It then extracts pitch correctly in voice sectors, interpolates wave forms, and extends the speech speed by maintaining the pitch at a fixed level. By controlling the soundless sections, the system extends intervals.

Superimposition through voice recognition. A continuous voice recognition system would automate superimposition of voices enunciated by an indefinite number of speakers in broadcast programs (news, for example).

Listeners base recognition on "knowledge such as grammar." The system recognizes the vowel portion of voice inputs, compares the consonants of each candidate, and outputs the most appropriate recognition results. In concrete terms, the voice to be recognized becomes a point of position within the vector space expressing the frequency characteristics. Vowel recognition occurs according to the distance from the representative point consonants and their proximity to the statistical model (hidden Markov model). In cases of limited usage, the system recognizes units of sentences (units of commonly used clauses) from continuous voice samples. Researchers are currently experimenting with a small-scale system that uses several parallel microprocessors.

Recording and tape technology

Recording technology represents another important area affecting overall program quality. Both video tape recorders and tapes themselves can determine the overall excellence of Hivision.

Reproduction of recordings on high-density vertical magnetic tapes. Smaller and longer duration digital video tape recorders demand the availability of high-performance tapes. Researchers have developed the high-performance cobalt-chromium-tantalum vertical magnetic tape for this purpose. When compared to the conventional stretched magnetic recording, this tape shows superior performance in reproduction of lengthy shortwave recordings.

Linear scanning VTR. The recent increase in the amount of information required to record images ties directly to the development of higher resolution displays and higher quality processing. The desire for mobility coupled with the increase

fueled an interest in investigating improved recording densities and thinner tapes. Research has resulted in a new VTR head scanning method that makes the mechanical system compact and is suitable for ultra-thin tapes.

Rotating optical recording head. Optical magnetic tapes would permit high-density recording, thereby increasing recording capacity. Research continues on development of an optical magnetic tape for use as an ultra-large capacity optical memory for data preservation and 3D pictures.

Displays

When choosing Hivision over regular television, the consumer will often base the decision on the quality of the image displayed. NHK's work in this area has focussed on developing flat-surface televisions using several display technologies.

Hivision TV with a plasma display. Full enjoyment of Hivision video requires a large screen. The weight and volume constraints point to the so called "wall-type television" as most practical. Investigators have selected the plasma display (discharge plasma), which is relatively simple to enlarge, and are proceeding with development a wall-type television of over 50 inches.

High-definition EL panel display. Electro-luminescent panels possess superior characteristics, such as thinness, light weight, self-luminous, and wide field of view. In addition, these all-solid panels show promise as high-definition, flat-surface displays. Researchers have currently developed and test produced delicately detailed pitch display panels with high-intensity EL layers.

Miscellaneous

Many projects under investigation at NHK do not easily fit under a major category. Although their applications generally deal with television and broadcasting, these projects carry on almost independently.

Kite plane. When gathering data about natural and man-made disasters, aerial photographs provide great impact. A kite plane equipped with superior safety features allows photography under conditions normally unsafe for human pilots and photographers.

Liquid crystal lighting elements. Liquid crystal optical modulating elements possess superior characteristics for high rates of transillumination. These devices do not require deflector plates, have rapid response, and easily cover a wide area. Investigators have developed heat resistant liquid crystal optical modulating elements (with movements possible at temperatures in excess of 160°C).

Radio disturbance reduction technology. Pulse static, a primary cause of TV and radio disturbances, originates primarily from degraded insulation of transmission and distribution lines and thermostats in old refrigerators and freezers. However, this static occurs in an irregular, mainly intermittent

manner, making identification of its origin difficult.

Researchers have solved this problem by developing a detector that measures instantaneous pulse static as well as continuous static waves and their direction.

Luminous multiporous silicon. Investigators are conducting basic research on this luminous element. The topics under study include the effects of the microscopic size of multiporous layers, the effects of its quantum size, and luminance caused by chemical compounds on the surface.

Virtual reality systems for the auditorium. Acoustic design of concert halls is often a hit-or-miss proposition. Its effectiveness would increase with a system that allowed the designer to hear the results of design changes. Researchers are developing a virtual reality system that would predict simply, conveniently, and quickly the acoustics of a hall by using reflective sound data reproduced by a computer based on design blueprints.

Acknowledgment

I am very grateful to Frank Nagashima for translating most of this material, which was made available to us only in Japanese.

Questions regarding this column can be addressed via e-mail to David K. Kahaner, US Office of Naval Research, Far East, at kahaner@cs.titech.ac.jp.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 171

Medium 172

High 173



The PowerPC 601 Microprocessor

The PowerPC 601 microprocessor is the first of a family of processors based on the PowerPC architecture. The general-purpose processor contains a 32-Kbyte cache and a superscalar machine organization that allows dispatch and execution of up to three instructions each clock cycle. The bus interface and storage control mechanism can be configured for a wide range of system designs: low-cost desktop personal computers to high-performance multi-processor systems.

Michael C. Becker

Michael S. Allen

Motorola, Somerset

Charles R. Moore

John S. Muhich

David P. Tuttle

IBM Austin, Somerset

The PowerPC 601 microprocessor is the first implementation of the PowerPC architecture developed by IBM Corp. and Motorola, Inc. at the Somerset facility in Austin, Texas. Together with Apple Computer, Inc., the alliance is developing a family of processors that addresses computer markets requiring high performance, low-power consumption, and outstanding price/performance.

The 601 microprocessor project achieved three significant goals. The first was to bring a PowerPC processor into the marketplace quickly. The alliance teams (referred to as "we" after this for brevity) achieved this by leveraging existing IBM and Motorola technology. IBM's POWER (Performance Optimized with Enhanced RISC) architecture¹ formed the basis of the PowerPC architecture, and the RISC Single Chip (RSC) processor² the base for the 601 microprocessor. Motorola contributed various architectural considerations that helped to form the PowerPC architecture. In addition, the bus interface from the Motorola 88110 microprocessor³ served as a basis for the PowerPC 601 bus interface.

The second goal was to offer competitive performance at a low cost. We enhanced the base RSC design to allow more concurrent instruction execution and higher frequency operation. Integrating the processor onto a relatively small die

and selecting an economical package minimized the cost.

The final goal was to offer capabilities suitable for a wide range of system design points. We achieved this by extending the versatile Motorola 88110 interface to support the PowerPC architecture, advanced protocol operations, and enhanced multiprocessing support.

Architecture overview

The PowerPC architecture is a third-generation RISC architecture designed for diverse computing requirements.⁴ The result is a powerful architecture that embraces fundamental concepts of simplicity and general applicability. It is also extensible to advanced techniques that will allow it to be used in future generations of PowerPC microprocessors.

Architecture goals. We felt it important to maintain an application binary interface (ABI) that was compatible with IBM's POWER architecture. Such compatibility allows PowerPC-based machines to exploit the existing application base of the RISC System/6000 machines. To that end, we adopted the POWER user-level instruction set and programming model as a starting point for the architecture. Although some instructions were ultimately added and others deleted, these changes can be managed by compilers and operating systems.

We also wanted to simplify the architecture to ease unnecessary implementation constraints. This flexibility allows optimizations appropriate for specific market targets. In addition, the simplifications permit smaller, faster, and more aggressive superscalar implementations.

A third objective was to provide support for a wide range of uniprocessor and multiprocessor system configurations. Recognizing key abstractions of the storage hierarchy and defining the storage control architecture to allow effective management of these abstractions achieved this goal. The architecture also allows storage references to follow either a big-endian or a little-endian byte-ordering convention to support different operating system needs.

The final architecture goal was to define 64-bit extensions that allow upward compatibility of 32-bit applications. We defined 64-bit instruction operation and 64-bit memory management as a logical extension to the 32-bit execution model. To allow flexibility, each implementation can either comply with the base 32-bit PowerPC architecture or the extended 64-bit architecture.

Autonomous execution units. One fundamental concept of the architecture involves the partitioning of the architecture. The specification divides the execution of instructions into three logically distinct processing units: branch, fixed point, and floating point. The units are loosely coupled so instruction execution can occur concurrently. Note that this is an architectural partitioning that does not impose implementation constraints. For example, the architecture allows implementations to provide multiple copies of any of the units for added performance or to combine any of the units for more efficient silicon area use.

We structured the branch processor architecture to allow early handlings of branch instructions. Resources architecturally defined as part of the branch processor generate target addresses and evaluate branch conditions. This logical partitioning lets the branch processor completely remove branch instructions from the execution stream and execute them in parallel with operations occurring in the other functional units.

The branch processor architecture defines three user-accessible branch control registers and several forms of branch instructions. The link register in conjunction with certain branch instructions provides efficient subroutine linkage. The count register acts with conditional branch instructions to construct iterating loops. The condition register contains eight 4-bit condition fields, which are set by a wide range of instructions. Branch instructions can be conditional on a bit in the condition register, conditional on the state of the count register, conditional on both registers, or simply unconditional. The branch target address can be absolute, program counter relative, or indirect from either the link register or the count register. We also defined a set of instructions to allow logical operations and movement of fields in the condition register.

The PowerPC fixed-point architecture defines 32 general-purpose registers and a rich set of computational, logical, shift, and storage access instructions. A full complement of specified add, subtract, multiply, divide, and logical instructions operate on either 32-bit or 64-bit operands. In addition, the computational model allows the construction of extended-precision arithmetic operations from these base operations. The architecture also specifies a powerful set of rotate-with-mask and shift instructions.

The fixed-point processor controls addressing for all storage access instructions. Addresses are generated either from a base register, plus a displacement from the instruction, or from a base register, plus the value in an indexing register. These instructions allow byte, half-word, word, and double-word access to and from storage, and may also specify an automatic update of the base address register.

The architecture also specifies instructions for moving larger blocks of data between the registers and storage. The load/store multiple instructions are useful for subroutine linkage, and the load/store string instructions are useful for movement of byte strings.

The floating-point architecture defines 32 double-precision floating-point registers and a computational model that conforms to the IEEE Standard for Binary Floating-Point Arithmetic.⁵ Computational instructions and storage access instructions support both single- and double-precision operands. In addition to the normal set of computational instructions, the architecture provides support for a powerful set of floating-point multiply-accumulate instructions.

The architecture allows implementation flexibility in the floating-point processor. Sophisticated implementations can commit the entire specification into hardware for high performance, while low-cost implementations could, for example, optimize the machine organization for efficient single-precision support. It is also possible for implementations to trap instructions and have software provide the required functionality.

Storage control. The architecture provides a robust storage control structure, which includes definitions for memory management, caching, and memory operations. The 32-bit architecture supports a 52-bit virtual address and a 32-bit real address. The 64-bit architecture supports an 80-bit virtual address and a 64-bit real address. Address translation occurs in two steps. A segmentation process translates the effective address into the virtual address, then a paging process translates the virtual address into a real address. Page translation is implemented through a memory-resident hashed page table. The page table is commonly cached in a translation look-aside buffer, so the architecture provides instructions to control the TLB. We fixed the page size at 4 Kbytes. In addition to the segmentation and paging mechanisms, the architecture also defines a block address translation mechanism that allows translation of blocks that can vary in size from 128 Kbytes to 256 Mbytes.

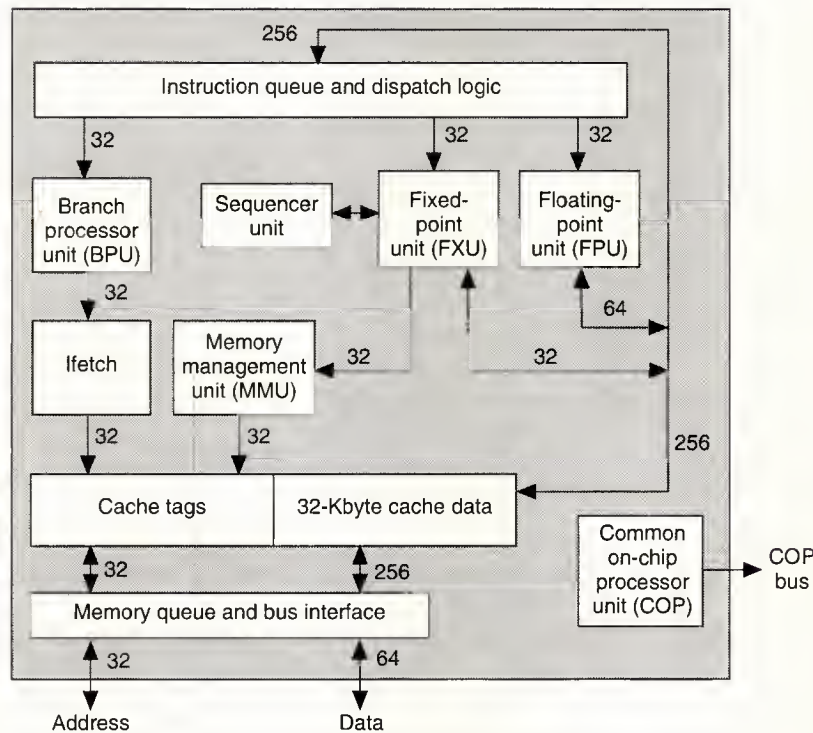


Figure 1. PowerPC 601 microprocessor block diagram.

The architecture specifies a generalized storage coherency model adaptable to a variety of coherency protocols. This model guarantees a certain behavior of storage references so multiple processors and devices can share memory. The specification gives implementations the freedom to define the particular protocols used to enforce coherency. A reservation mechanism coordinates with the coherency scheme to permit implementation of shared-memory synchronization primitives. Storage attributes are set at both page and block levels, and allow specification of cachable, write-through, memory coherent, and guarded storage. The architecture also defines cache control instructions to allow software-directed coherency and application tuning. In particular, it specifies instructions for data cache invalidate, instruction cache invalidate, data cache flush, data cache store, data cache allocate with zeroes, and data cache touch.

The PowerPC architecture defines a relaxed storage consistency model. This model guarantees sequential consistency as viewed by a program but allows considerable implementation flexibility in the ordering of memory references. This weak consistency model allows the processor to achieve higher levels of performance through speculative accesses, runtime reordering of storage accesses, and optimized use of the memory resource. The architecture also defines a set of

synchronization and barrier instructions to allow software to achieve the effect of a stronger ordering model. These instructions are particularly useful for memory-mapped I/O applications.

The combination of the large address spaces, versatile memory management structures, and generalized storage coherency and consistency models will help to ensure the longevity of the architecture.

Machine organization

The PowerPC 601 microprocessor is a 32-bit implementation of the PowerPC architecture. The superscalar machine organization can dispatch up to three instructions each cycle. The 601 microprocessor also benefits from instruction pipelining, which helps to improve its clock rate. Figures 1 and 2 illustrate the block diagram of the 601 microprocessor, and Figure 3 shows the pipeline structures.⁶

The instruction queue and dispatch logic buffers instructions from the cache and dispatches up to three instructions each cycle—one each to the fixed-point unit (FXU), the floating-point unit (FPU), and the branch processing unit (BPU). The FXU communicates with the sequencer unit for handling infrequently used instructions and certain control-intensive tasks. In addition, the FXU interfaces with the memory management unit (MMU) for cache accesses. The 32-Kbyte, unified cache provides a 32-bit interface to the FXU, a 64-bit interface to the FPU, and a 256-bit interface to both the instruction queue and the memory queue. The chip interface includes a 32-bit address bus and a 64-bit data bus. In addition, the chip supports an asynchronous serial port, the common on-chip processor (COP) bus, to support debugging and test features.

Instruction queue and dispatch unit. The 601 microprocessor contains an eight-entry instruction queue for prefetched instructions. The queue is fed by an eight-word bus from the cache. During each cycle, the dispatch logic considers the bottom four entries of the instruction queue and dispatches up to three instructions (Figure 4). The queue supports the full range of possible shift amounts with no instruction alignment restrictions on the program.

The queue positions from which instructions can be dispatched vary for each function unit. Branch instructions and most floating-point instructions can be dispatched from any of the bottom four entries of the instruction queue. Fixed-point instructions are always dispatched from the bottom queue entry. Floating-point stores are dispatched to both the

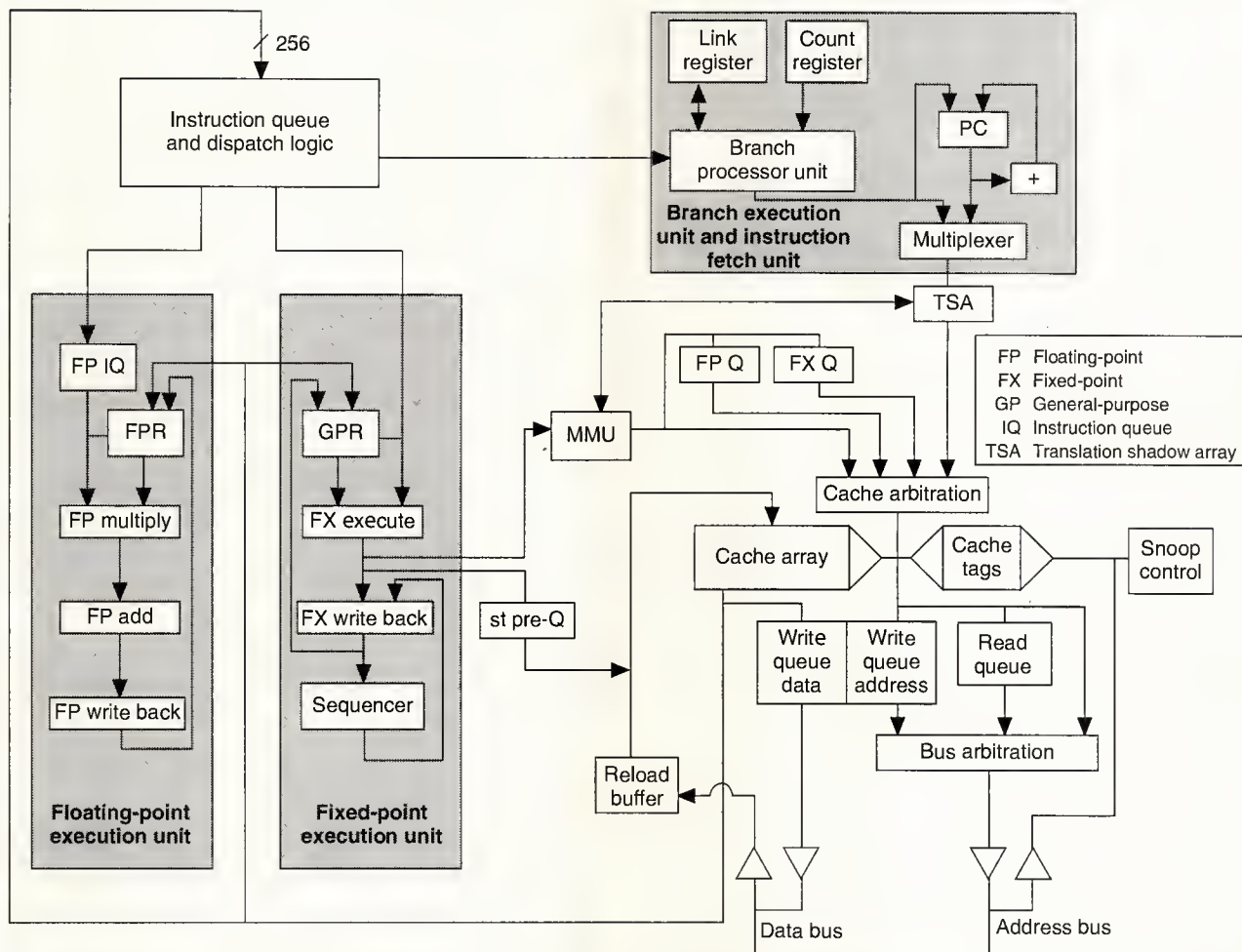


Figure 2. Internal dataflow diagram.

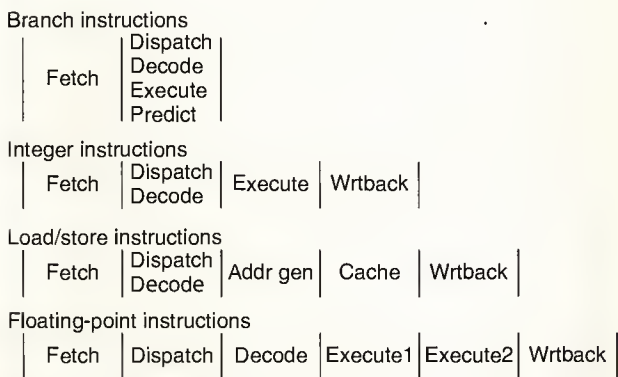


Figure 3. Pipeline description.

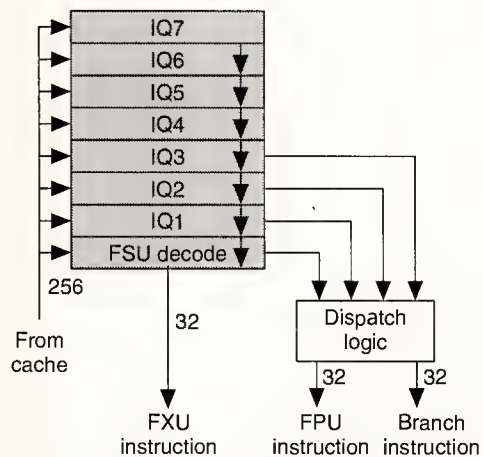


Figure 4. Instruction queue and dispatch logic.

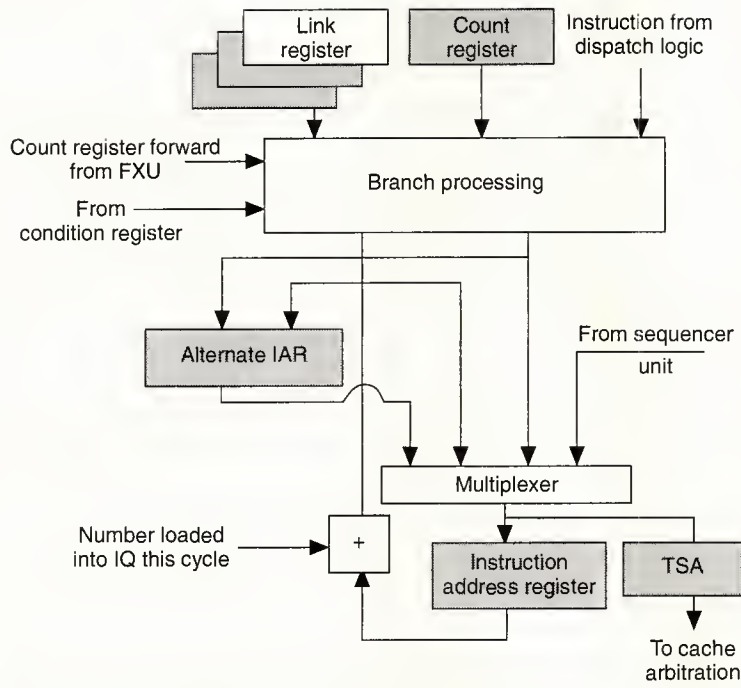


Figure 5. Instruction fetch and branch processor units.

FXU (for address generation) and the FPU (for data sourcing). Floating-point loads are dispatched only to the FXU, but the FPU is made aware of any dependencies, and data transfers directly to the FPU.

The 601 microprocessor permits out-of-order dispatch such that branch instructions and most floating-point instructions can be dispatched and removed from the queue even if a preceding fixed-point instruction is interlocked. These folded instructions execute concurrently and do not occupy a position in the fixed-point pipeline. A tagging and counting mechanism preserves the program order completion of these instructions. Out-of-order dispatch is more complex to implement, but it allows the 601 microprocessor to expose subsequent branches earlier, which can reduce the dispatch stalls that may otherwise result.

Instruction fetch unit. This unit coordinates instruction fetching from the cache. Several different sources in the 601 microprocessor generate instruction fetch addresses. The branch processor provides the address that results from branch instructions. The sequencer unit provides addresses associated with interrupts and other synchronizing events. The instruction fetch unit itself generates the next sequential address in the event that no branch or interrupt has occurred. During each cycle, the appropriate address is selected, translated, and forwarded to the cache arbitration logic for consideration to access the cache. The instruction queue and dis-

patch logic accepts and processes instructions fetched from the cache.

The instruction fetcher also provides fast address translation of instruction fetch addresses through the translation shadow array. The TSA automatically keeps track of the four most recently used instruction address translations and provides fully associative comparison in response to the address generation of any instruction fetch. The TSA supports both page- and block-oriented address translations. In the event of a miss in the TSA, the instruction fetcher arbitrates for access to the 601's primary MMU for translation, and then updates the TSA based on a least recently used replacement (LRU) algorithm.

Branch processor unit. The BPU (Figure 5) executes branch instructions in its two-stage pipeline (Figure 3). The 601 microprocessor dispatches, decodes, evaluates, and, if necessary, predicts the direction of branch instructions in one cycle. On the next cycle, the resulting fetch can access new instructions from the cache. This allows the processor to quickly react to branches detected in the instruction stream and to reduce the latency of subsequent instructions.

Unconditional and count register dependent branches execute during the same cycle that they are dispatched. As a result, they present no delay in the instruction dispatch stream to the FXU or the FPU and effectively execute in zero cycles. Conditional branches dependent on the condition register can either be resolved or unresolved at the time of dispatch. Each of the eight fields of the condition register has a set of associated interlocks activated by instructions that update that field. If a conditional branch is dependent upon a noninterlocked condition register field, the branch can be resolved immediately. In this case, the condition is evaluated based on the contents of the condition register, and the branch presents no delay in the instruction dispatch stream to the FXU or FPU.

Branches conditional on an interlocked condition register field are unresolved. In these cases, the 601 microprocessor employs a static branch prediction algorithm to predict the direction of the unresolved branch. The 601 microprocessor algorithm predicts that the branch will be taken if the displacement of the target address is negative, and predicts it will not be taken if it is positive. As an aid for compiler-directed prediction, a bit in the branch instruction opcode allows this prediction scheme to be reversed. Instructions fetched on behalf of a predicted conditional branch can be conditionally dispatched, but they do not execute until the prediction has been validated.

Eventually, condition register interlocks are released, and the validity of the prediction can be checked. If the prediction was correct, the branch completes and all prefetched instructions are marked eligible for execution. In the event that the prediction was incorrect, any incorrectly prefetched instructions are discarded and fetching resumes down the correct path.

The 601 microprocessor has several features to speed up recovery after a misprediction. First, a fast alternate address restore mechanism saves the alternate path address and immediately redirects the instruction fetcher. Second, the FXU forwards the results of compare instructions directly to the branch processor as well as the condition register. This allows earlier resolution of unresolved conditions. Finally, the instruction queue implements a delayed purge mechanism, which retains any prefetched sequential instructions beyond the conditional branch until the prefetched predicted instructions are made available for loading into the instruction queue. In the event that the condition is resolved before the predicted instructions are available (the likelihood of which is increased by the optimizations for the compare instructions in the FXU), and the prefetch was incorrectly predicted down the taken path, dispatch simply continues down the sequential stream. This stream consists of the instructions remaining in the instruction queue.

Fixed-point execution unit. The FXU receives instructions from the instruction dispatcher. In general, the FXU serves as the master pipeline, managing the synchronization control required to achieve precise exceptions. The instructions execute in a four-stage pipeline (fetch, decode, execute, write back; Figure 3), and the hardware interlocks all hazards (Figure 6). All architecture instructions process in hardware, and most execute in a fully pipelined manner. Some of the arithmetic (multiply and divide) and multiple word storage operations require several cycles in the execute stage. To improve the effective execution latency, the FXU forwards any register-dependent results to their respective function units at the same time the data is written back to the register file. In addition, although the differences between the POWER architecture and the PowerPC architecture at the user level are small, the 601 microprocessor also provides hardware support for all POWER architecture user mode instructions.

The FXU handles the address generation portion of all load and store instructions. This includes the floating-point loads and stores, although in some cases, the instructions also progress through the floating-point pipeline. A dependent operation following a load instruction causes a one-cycle load

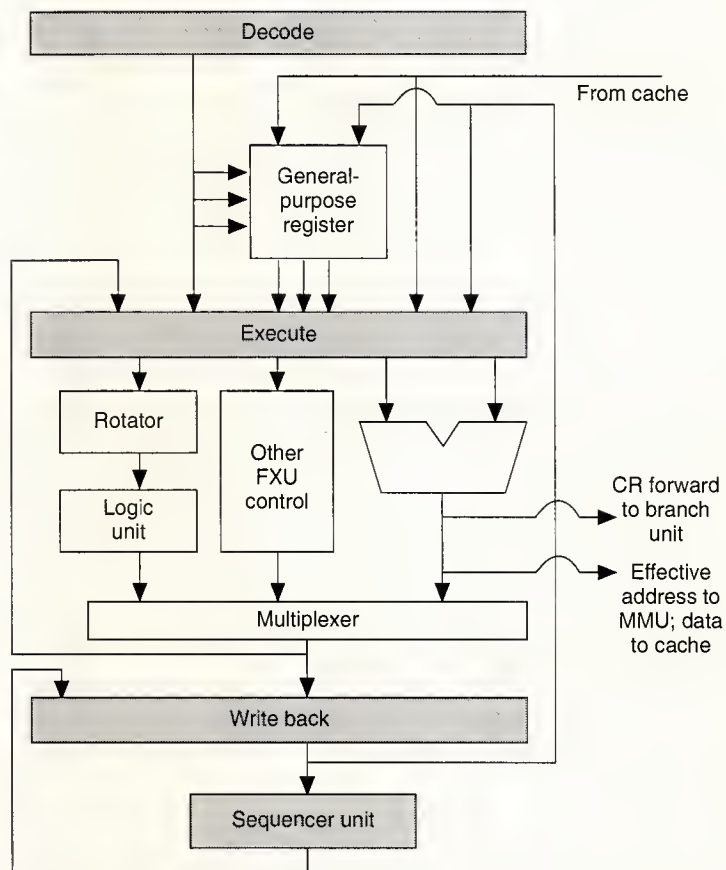


Figure 6. Fixed-point execution unit.

delay slot. The FXU sequences the execution of the load/store multiple instructions and the load/store string instructions. The hardware handles most forms of unaligned addresses; however, in some cases, multiple accesses from the cache are necessary.

The FXU features a fast execution of compare instructions and the ability to forward results directly to the branch processing unit. As discussed previously, this allows efficient handling of branches dependent on the result of the compare instruction.

Floating-point execution unit. The FPU (Figure 7, next page) complies with the IEEE-754 standard for both single- and double-precision floating-point arithmetic operations. The FPU supports the compound multiply-add operations that are defined in the PowerPC architecture. Furthermore, the hardware automatically handles all floating-point data types including denormalized numbers, NaNs, QNaNs, and infinities. The FPU pipeline has six stages (Figure 3). The decode stage contains 32 double-word registers, the instruction de-

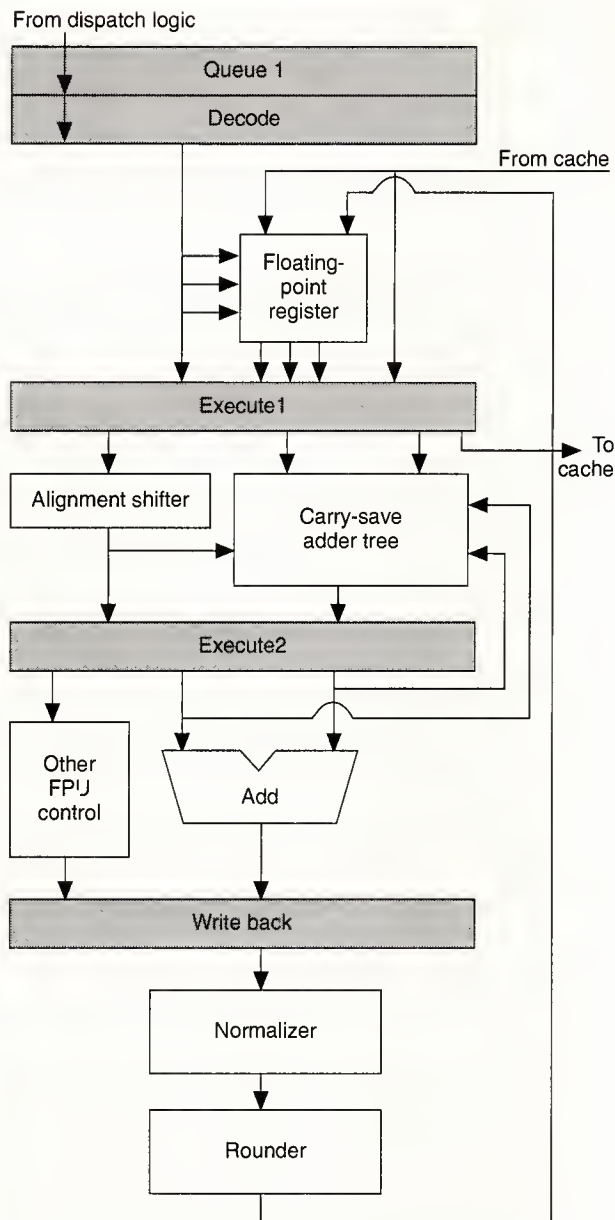


Figure 7. Floating-point execution unit.

code logic, and the main pipeline control for the FPU. The execute1 (or multiply) stage contains the Booth encoder and carry-save adder tree, and an alignment shifter for the second operand. The execute2 (or add) stage accepts the sum and carry values from the previous stage, and produces a single intermediate result. Finally, the write-back stage rounds and normalizes results, and updates the register.

The FPU is optimized so that most single-precision operations (all except divide) and most double-precision operations (not involving multiplication or division) that involve operands with normal data types are fully pipelined. This achieves a four-cycle latency and one-cycle throughput. Double-precision operations involving multiplication are obtained by double pumping each pipeline stage to achieve a five-cycle latency and a two-cycle throughput (Figure 3). Division is performed using a 2-bit nonrestoring division algorithm, which takes 17 cycles for single precision and 31 cycles for double precision.

The FPU receives instructions from the instruction dispatch unit. In many floating-point-intensive applications, it is possible that the FPU will lag behind the full dispatch rate of the dispatch unit. To alleviate the adverse effects of this, the FPU maintains an additional two-entry instruction queue. This allows the dispatch logic to remove floating-point instructions from the primary instruction queue and expose subsequent fixed-point and branch instructions earlier.

The FPU operates independently from the FXU and can concurrently execute floating-point instructions. A carefully tuned synchronization scheme allows these execution units to progress independently while maintaining precise exceptions. These pipelines cooperate in the execution of floating-point storage access instructions but are not required to proceed in lock step. For example, the FXU can process the addressing portion of a floating-point store operation and pass it onto the cache unit without waiting for the FPU to produce the store data. The cache unit will complete the store operation once the FPU provides the data.

The 601 microprocessor provides a mode that forces enabled floating-point exceptions to be reported in a precise manner. This useful mode also serves as a software debugging aid for floating-point programs that produce degenerate results.

Sequencer unit. This unit is essentially an embedded support processor that assists the core CPU in handling many of the algorithmic functions of the PowerPC architecture. It sequences the power on reset functions, which include array self-testing for the cache and array initialization for all other storage cells on the chip. It maintains an on-chip real-time clock and the system decremter function. In addition, it handles the recording and sequencing of interrupts, context-synchronizing events, and errors.

The sequencer also walks through the hashed page table for address translations that miss in the TLB, and, as appropriate, it updates the storage access recording bits. In addition, to minimize the chip area overhead, several of the less frequently used system control registers are physically implemented in a RAM structure within the sequencer. As a result, the instructions that operate with these registers are passed from the FXU to the sequencer for execution assist.

Memory management unit. The MMU (Figure 8) per-

forms the virtual-to-real address translation for load and store instructions. In addition, it acts as a backup for instruction fetch address translations that miss in the instruction fetcher's TSA. The MMU is tightly coupled to the FXU so that address translation can occur during the execute phase of the FXU pipeline. A cache access occurs the following cycle using the translated address. Arbitration logic determines whether the FXU or the instruction fetcher has use of the MMU during a particular cycle. The instruction fetcher's use of the MMU should be infrequent since the hit rate of the TSA is high for most applications.

The MMU provides support for segment, page, and block translations. Address translation begins with indexing into one of 16 segment registers. Control bits in the segment register control the mapping of an address to the real address space, the I/O space, or the virtual space. Virtual addresses are further translated by page or block translations mechanisms.

The PowerPC architecture specifies a hashed page table structure and a fixed-page size of 4 Kbytes. Portions of the virtual address are used to hash into the page table entries, which are subsequently examined for matching virtual page numbers. A matching entry produces a real page number, which can then be used to access real memory. If no match is found after 16 attempts, a page fault exception is taken. In an effort to speed up the page-based address translation, the 601 microprocessor provides a 256-entry, two-way set-associative TLB cache that is used for both instructions and data. Updates to the TLB and maintenance of the storage access recording (reference and change) bits are handled in hardware.

The BAT, or block address translation, mechanism allows for up to four block mappings that can vary in size from 128 Kbytes to 8 Mbytes. This is useful for handling large sections of memory that are not subject to demand paging but still require address translation and protection. Proper use of the block translation registers can reduce TLB thrashing and increase overall processor performance.

Cache structure. The 601 microprocessor provides a unified 32-Kbyte, eight-way set-associative cache (Figure 9, next page). Each 64-byte line holds two 32-byte sectors; cache indexing and cache tag comparisons use the real (physical) address. In general, the cache makes use of a store-in (or copy-back) policy to store data. However, the cache allows page- and block-level control of cachability, write through, and coherency (via the MMU). An LRU cache replacement

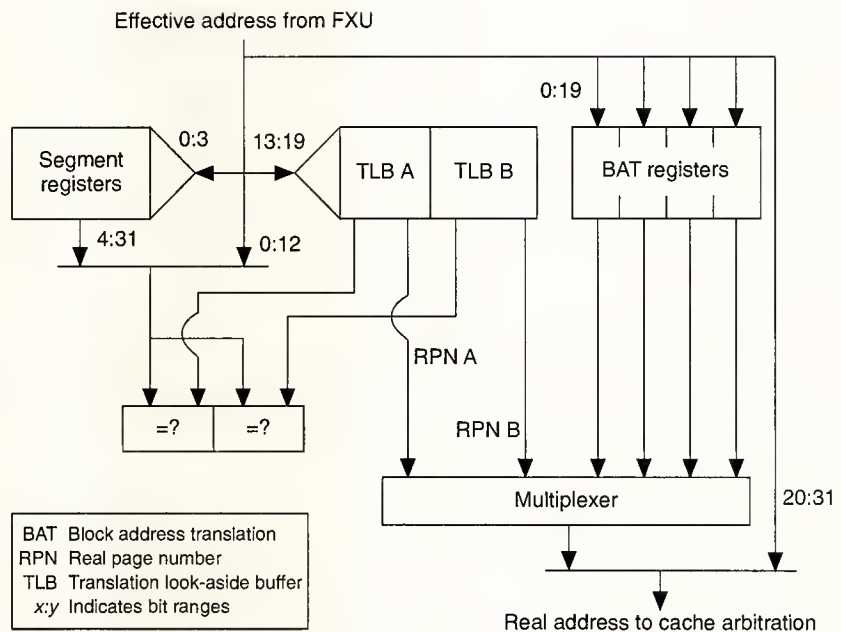


Figure 8. Memory management unit.

algorithm is used, and coherency is maintained on a sector basis using the four-state MESI (modified, exclusive, shared, invalid) cache coherency protocol.⁷

In a unified cache implementation, it is important to make the most effective use of the available cache bandwidth. The 601 microprocessor employs several techniques to achieve this. First, the cache interface to the fetch unit and the memory queue is 8 words wide. At 66 MHz, the data rate for this interface is 2.1 Gbytes/s. Although all eight words are not used every cycle, this bandwidth speeds sector cast-outs (cast-outs refer to modified data moves to memory), snoop pushes, and instruction fetching. Second, the cache is capable of a complete read-modify-write every cycle, which allows the 601 microprocessor to process store operations in one cache cycle. Third, a balanced arbitration scheme prioritizes the various cache access requests that can occur each cycle. To prevent cache arbitration from stalling the execution units, operation queueing is provided above the cache for fixed-point and floating-point storage operations. Finally, the cache is a nonblocking design, and a memory queue sits below the cache to hold pending storage operations.

The 601 microprocessor cache supports all PowerPC cache control operations. These are useful for tuning applications in which memory performance can be improved by explicit cache management, and for systems that perform software-assisted coherency schemes.

The cache is a full-custom design that exploits leading-

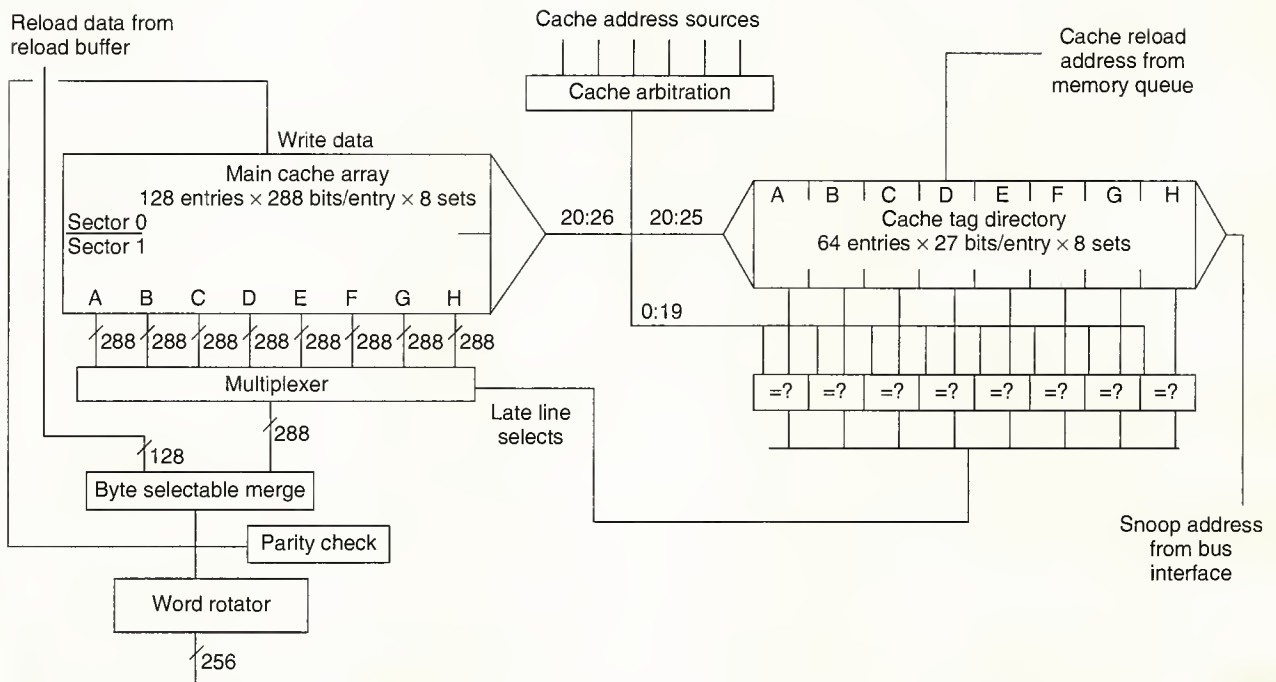


Figure 9. Cache organization.

edge circuit design techniques to achieve the required density and performance characteristics. The cache uses an electrically stable six-transistor static RAM cell and is protected with byte parity for enhanced error detection and reliability. A deterministic self-test procedure checks the cache as part of the power-on reset process.

Memory queue. The memory queue cooperates with the cache and the bus interface unit to queue memory transactions. It decouples the cache from the bus interface, which allows the cache to continue providing data for instruction fetches and load/store operations, independently of the bus transactions still in the queue.

The memory queue is made up of a two-entry read-address queue and a three-entry write-address/data queue. The write queue holds up to three 32-byte sectors of dirty (modified) data. The read queue, the write queue, and in some cases the cache itself, arbitrate for access to the 601 bus. The memory queue maintains full hardware-enforced coherency to the processor, cache, and memory. Additionally, the memory queue supports a dynamic reload feature, which permits background fetches for the other sector of a cache line after the critical sector has been successfully loaded. When enabled, this feature provides an automated cache prefetching capability for both instructions and data.

Bus interface unit. The 601 BIU translates memory requests from the memory queue and the cache into transac-

tions on the bus interface. It has a 32-bit address bus and a 64-bit data bus and uses an efficient bursting protocol to achieve high data throughput. The TTL- or CMOS-compatible interface runs at the processor clock rate or any integer multiple of the processor clock period. Figure 10 is a block diagram of the BIU with the memory queue.

The BIU employs a bursting protocol that uses one address to transfer four beats of data (8 bytes each). While the theoretical limit of data bus throughput is the product of bus width and bus frequency, the realizable bandwidth is limited by factors such as interface technology, bus arbitration overhead, address transfer overhead, memory latency, and interprocessor communication. In the 601 microprocessor, consecutive burst transfers require one-cycle separation for an effective throughput of 422 Mbytes/s at 66 MHz. The protocol specifies independent operation of the address and data buses to allow the bus to approach its bandwidth limit.

A simple bus protocol employs a tenured bus in which the memory latency governs the use of the address, as shown in the first example in Figure 11. If the memory latency is long, the actual data bus bandwidth can be significantly degraded. To overcome this problem, the 601 address and data buses operate independently, and new addresses can be launched before the preceding data transfer has completed, as shown in the address pipelining example. The 601 microprocessor itself can pipeline up to two outstanding addresses on the bus. In multiprocessing

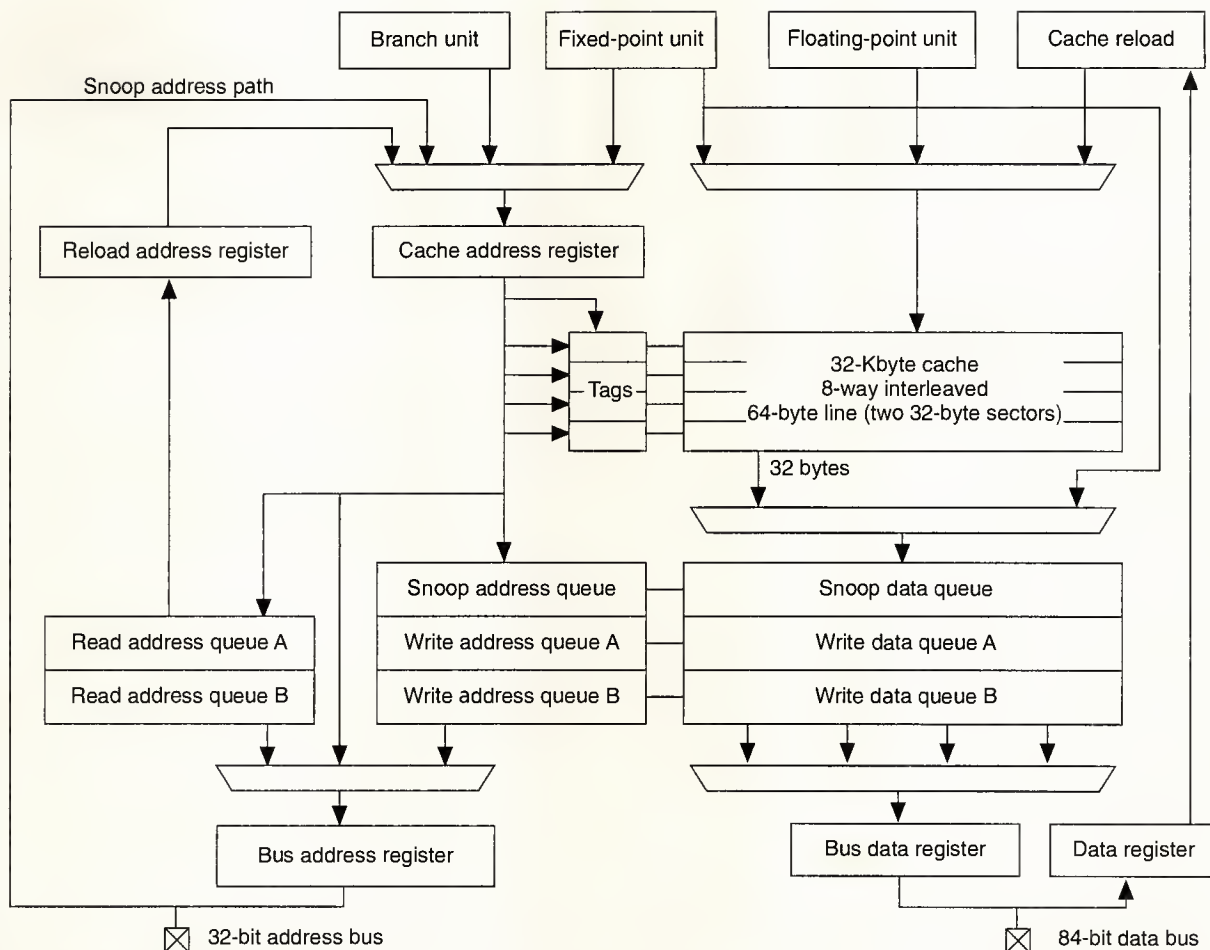


Figure 10. Memory path block diagram.

configurations, the protocol can support out-of-order completion of transactions and a greater number of outstanding addresses. This further improves bus efficiency in systems with variable memory latency (system I/O vs. memory, cache memory, banked memory designs).⁸

The interface protocol uses three phases to control the address and data buses. Figure 12 (next page) is a diagram of a basic 601 bus transaction illustrating the three phases. The following outlines the basic operation of each of the phases defined for the protocol:

- **Arbitration.** Arbitration signals request and grant ownership separately for the address bus and the data bus.
- **Transfer.** The address bus and the transfer

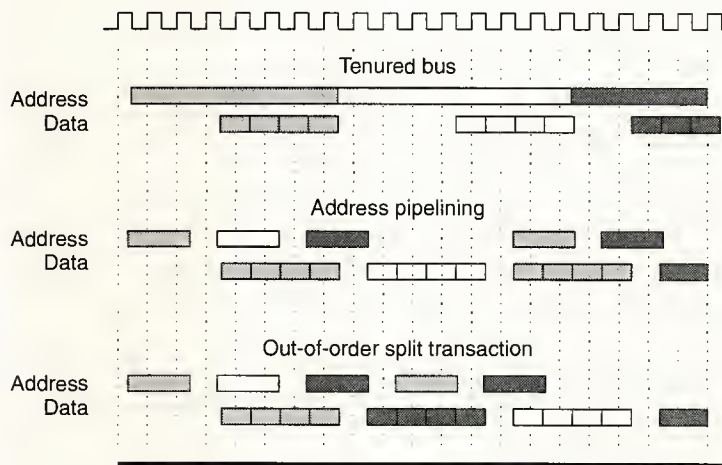


Figure 11. Bus optimizations.

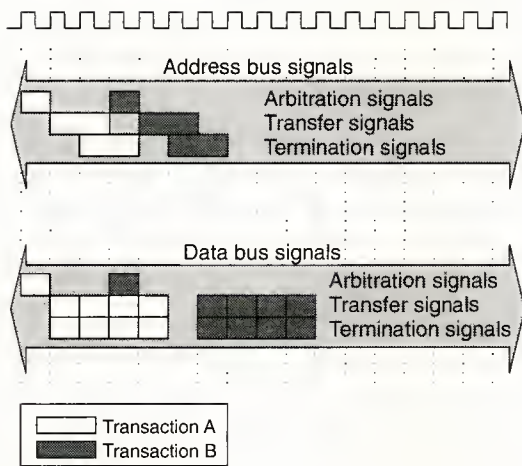


Figure 12. Bus transfer protocol.

attribute signals are driven to designate the location, type, and size of the transfer. Transfers include burst, single-beat, reservation, broadcast cache control, and broadcast TLB control operations. Either the 601 microprocessor or the target of the transfer drives the data bus signals.

- **Termination.** Termination signals conclude the bus tenure separately for each bus.

Reservation transfers are bus operations initiated by the reservation mechanism and are used to achieve shared-memory synchronization. The program-initiated cache and TLB broadcast operations provide remote control of caches and TLBs in other processors on the bus. These address-only transactions are illustrated in the middle of the address pipelining example (Figure 11). Since the data bus is typically more heavily used than the address bus, these address-only operations can transparently use the spare bandwidth of the address bus to data bus operations.

The 601 microprocessor maximizes opportunities to use the bus resource by prioritization of internal requests for access to the bus. As discussed earlier, the PowerPC architecture defines a weak consistency model, which, for most instruction streams, allows memory operations to complete in a different order than programmed. This has two positive effects on bus throughput. By allowing the longer latency read operations to start on the bus before the older (with respect to program order) store operations, the hardware reduces the latency of the read operation. Also, placing the dirty cache sector cast-outs and the dynamic reload requests lowest in priority allows these opportunistic operations to attempt to fill the gaps between higher priority bus operations.

The 601 microprocessor minimizes memory access latency through bus parking, speculative forwarding of data, and a

short cache-to-memory path. Bus parking eliminates an arbitration cycle by allowing the external arbiter to grant the 601 microprocessor the bus before the 601 makes a request. Speculative data forwarding is permitted through the definition of a data retry function that can be used to invalidate the double word transferred on the previous bus clock cycle. This is useful for systems employing secondary cache systems that forward data prior to the tag comparison and for correction algorithms that protect memory integrity. The 601 microprocessor reflects substantial design effort to minimize latency from cache miss detection to the return of instructions or data to the cache.

Bus timing example. System-level optimizations are critical to performance, particularly as the processor core often operates at a multiple of the bus clock frequency. Reducing memory latency at the system bus level by a bus clock cycle can potentially reduce latency as seen by the processor by several clock cycles. The 601's innovations to reduce latency are best shown by a best case example using the following assumptions:

- the bus is running at the same frequency as the processor;
- there are no higher priority operations queued;
- the 601 microprocessor is parked on the bus and does not need to assert a request; and
- the 601 microprocessor is connected to a fast, direct-mapped second-level cache using the speculative data-forwarding capability of the 601 interface.

In clock cycle 1, the physical address is driven to the cache directory, the two read queues, and the bus address register in parallel. One of the read queues is loaded, as well as the bus address register. Also in cycle 1, the result of the cache directory indicates a cache miss; therefore, the bus controller can drive the address to the bus on the following cycle. Since it has already been granted the address bus, the 601 microprocessor drives the address in clock cycle 2. (Figure 13).

The memory system responds with the first double word of data in clock cycle 3, and the subsequent double words of data in clock cycles 4, 5, and 6. The bus interface logic gathers two beats (four words) of data together before requesting access to the cache in clock cycles 5 and 7. The write back into the cache takes place late in clock cycle 6. The requested datum or instruction packet is simultaneously forwarded to the FXU, FPU, or branch unit in clock cycle 6.

Deadlock prevention in hierarchical buses. A special scenario arises in hierarchical bus environments that consist of a processor bus connected via a bridge component to an I/O bus (like EISA). If the processor and a master on the I/O bus simultaneously request data from the other bus, one of the requests cannot continue. A processor may have its request retried so that the I/O bus request can proceed (available on the 601 microprocessor). However, this solution may cause

difficulties with devices that should see an address only once for a single request. The 601 microprocessor overcomes this by allowing a processor request to be suspended after the address has been transferred, thus allowing the I/O bus to proceed and complete before the 601's transfer is completed.

Multiprocessing features. The 601 microprocessor maintains coherency with a granularity of 32 bytes (equal to a cache sector). Granularity refers to the smallest block of memory that is acted on to maintain coherency. The 601 microprocessor also distinguishes between global and nonglobal transactions. Only global transactions are monitored (snooped) by other processors.

The bus supports the MESI coherency protocol used by the cache. Each bus transaction is classified, so the other processors maintain coherency. Other processors use this information to update the MESI state of their cache entries. Conversely, the 601 microprocessor must snoop the bus for addresses in its own cache and generate an appropriate snoop response (shared hit, dirty hit, miss).

A critical feature of the overall bus performance in multiprocessor systems is the snoop response latency, defined as the time the snoopers takes to respond to a valid address on the bus with a coherency reply. The 601 microprocessor has only two bus clocks of snoop response latency because of its dual-ported cache tags. Since this is uniform across all 601s in a multiprocessor system, the latency associated with the hardware-enforced coherency is deterministic and very short.

Sample 601 microprocessor system configurations. The 601's bus interface accommodates a wide range of system configurations. As an overview of the 601's bus capabilities, we show three systems in Figure 14 (next page). Figure 14a is a block diagram of a 601-based single-processor system. In this system, a simple arbiter can determine whether the memory bus is owned by the 601 microprocessor or by an expansion bridge to a system bus such as VME, NuBus, MicroChannel, and PCI⁹.

In Figure 14b, the 601 microprocessor interfaces to a second-level cache to provide improved uniprocessor performance. The 601 microprocessor provides a full complement of cache control signals to allow construction of a low-latency, high-performance cache. In this class of system the frame buffer will often attach to the memory bus for high performance.

The system in Figure 14c shows a multiprocessing system. Typical features of a 601-based multiprocessor might include

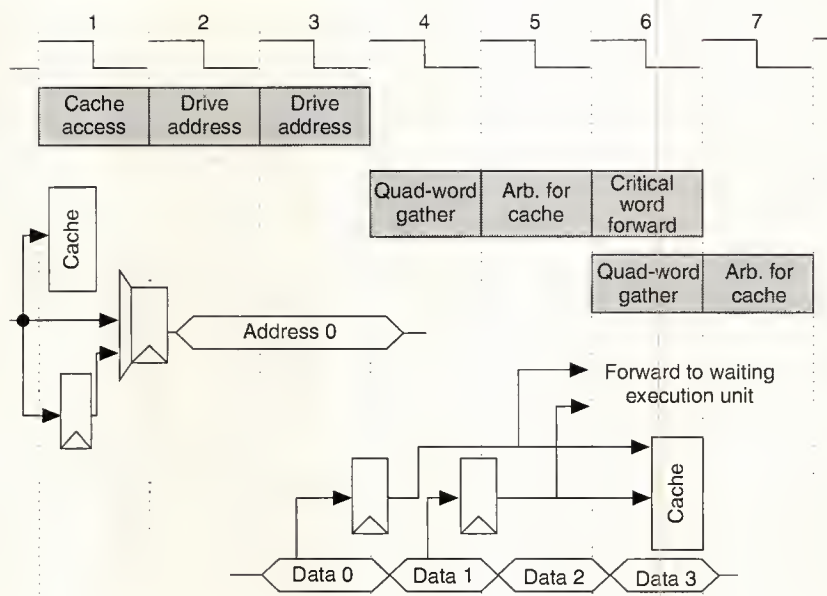


Figure 13. Cache miss timing.

a shared memory, a shared bus to facilitate hardware enforced coherency among a number of tightly coupled processors, a local cache system for improved processor and system performance, and cache control operations broadcast on the bus to allow processors and external hardware to control the local cache state. System designs using other interconnection topologies (data crossbars, hierarchical caches) can also be constructed.

Support for debugging and test

The common on-chip processor (COP) is the master control logic for the built-in self-test, debugging, and test features of the 601 chip. A simple serial command interface allows external devices to communicate with COP and initiate various actions. This command interface can be configured to be compatible with the IEEE standard¹⁰ for JTAG access to boundary scan. The COP features combine to create a powerful debugging environment with detailed visibility into the 601 microprocessor.

Chip and packaging technology

We implemented the 601 microprocessor in an IBM CMOS technology with 0.6- μ m minimum feature size and four levels of metal wiring. The 601 microprocessor is packaged in a 304-pin ceramic quad flat pack that is bonded using IBM's C4 solder ball technology. Table 1 summarizes other key physical characteristics of the 601 microprocessor.

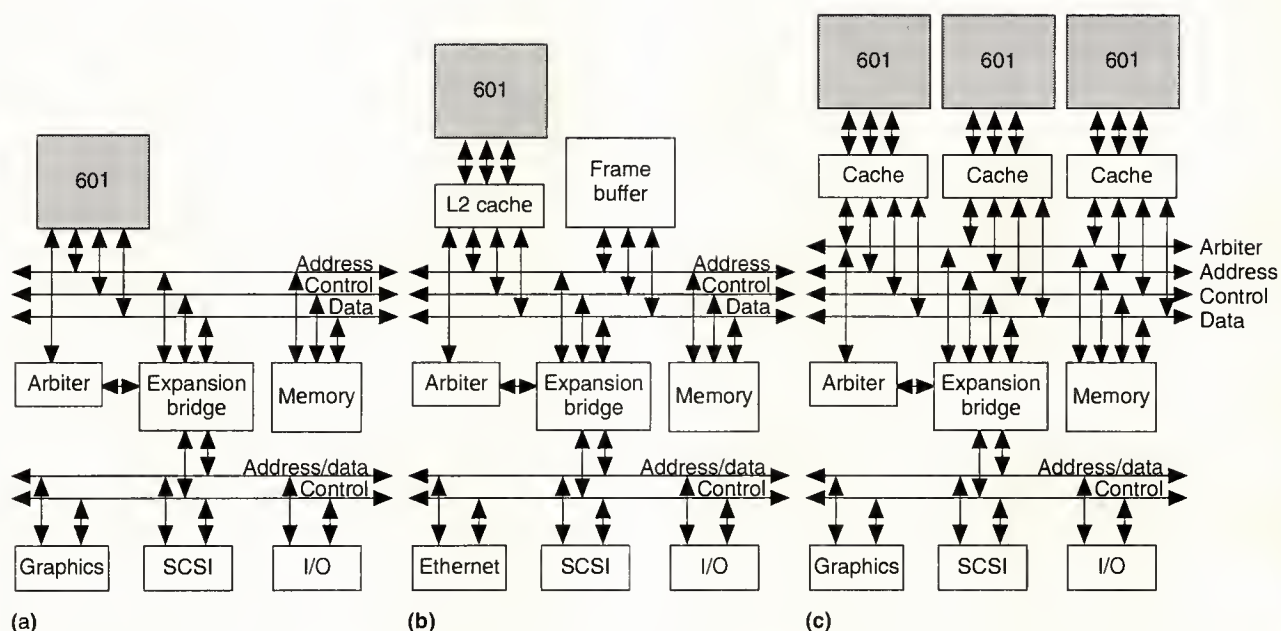


Figure 14. Typical 601-based systems: single processor (a), uniprocessor plus second-level cache (b), and multiprocessing system (c).

Table 1. Physical characteristics of the PowerPC 601 microprocessor.

Description	Characteristic
Technology	0.6- μ m CMOS; four metal levels
Die size	10.95 \times 10.95 mm
No. of transistors	2.8 million
Voltage	3.6
Power	6.5W at 50 MHz (typical)
Package	304-pin ceramic quad flat pack
Frequency	50 MHz and 66 MHz, initially
I/Os	184 signals; CMOS or TTL levels compatible

Performance

In addition to the key performance features of the 601 microprocessor, other factors, such as the operating system, memory characteristics, system support, and operating frequency, play an important role in the overall performance of the system. To establish a rough basis for comparison, Table 2 provides estimated SPEC performance for a hypothetical

Table 2. Power PC 601 microprocessor performance.

Frequency	SPECint92	SPECfp92
66 MHz	60*	80*
*Estimated performance		

system with a 66-MHz PowerPC 601 processor. Note that these are estimated projections and do not reflect the performance of a particular system.

THE DEVELOPMENT OF THE POWERPC 601 microprocessor was a unique project. It had aggressive functionality, performance, die size, quality, and schedule goals. It required the total dedication of many people from several divisions within IBM, from the Motorola staff, and from the Apple staff. The quick development of a close working relationship of these different groups is a testament to the dedication of the teams. This dedication is also reflected in the success of the 601 microprocessor.

The PowerPC 601 microprocessor successfully combines the architectural advantages of the PowerPC architecture, an efficient superscalar machine organization, and state-of-the-art CMOS technology to achieve an impressive level of functionality and performance. Critical performance features in the 601's machine organization include the branch unit, which can remove branches from the instruction stream, and the low-latency, high-density cache. In addition, the bus interface provides a flexible interface and a proven multiprocessor coherency protocol.

The 601's small die size and low-power consumption set new standards for microprocessors in its performance class. In the future the alliance will produce three processors that address low power, improved performance cost, and high-end performance. The combined efforts of IBM, Motorola, and Apple on the 601 microprocessor establish a firm foundation for the PowerPC family. ■

Acknowledgments

The authors acknowledge the contributions of the PowerPC Architecture Committee, logical designers, physical designers, verification and test engineers, packaging team, methodology support engineers, and all the other support staff who worked so hard and made many personal sacrifices. The result is a credit to their competence and dedication.

References

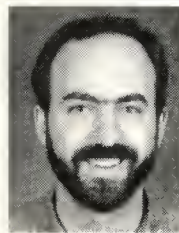
1. R.R. Oehler and R.D. Groves, "IBM RISC System/6000 Processor Architecture," *IBM J. Research and Development*, Vol. 34, No. 1, Jan. 1990, pp. 23-36.
2. C.R. Moore et al., "IBM Single-Chip RISC Processor (RSC)," *Proc. Int'l Conf. Computer Design*, Oct. 1992.
3. K. Diefendorff and M. Allen, "Organization of the Motorola 88110 Superscalar RISC Microprocessor," *IEEE Micro*, Vol. 12, No. 2, Apr. 1992, pp. 40-63.
4. E. Silha and G. Paap, "PowerPC: A Performance Architecture," *Proc. Compcon*, Feb. 1993, pp. 104-108.
5. *ANSI/IEEE 754-1985 Standard for Binary Floating-Point Arithmetic*, IEEE, Piscataway, N.J., 1985.
6. C.R. Moore, "The PowerPC 601 Microprocessor," *Proc. Compcon*, Feb. 1993, pp. 109-116.
7. *Future+, Logical Layer Specification, Draft 8.02 P896.1/D8.02*, IEEE Computer Society Press, Los Alamitos, Calif., Sept. 1989.
8. M. Allen and M. Becker, "Multiprocessing Aspects of the PowerPC 601 Microprocessor," *Proc. Compcon*, Feb. 1993, pp. 117-126.
9. *PCI Local Bus Specification, Revision 2.0*, PCI Special Interest Group, Hillsboro, Ore., Apr. 1993.
10. *IEEE Std 1149.1 Test Access Port and Boundary-Scan Architecture*, IEEE, Piscataway, N.J., 1990.



Charles R. Moore, an advisory engineer in the Advanced Workstation and Systems Division (AWS) of the IBM Corporation, is currently assigned to Somerset's High End Processor Development Group. He is the co-lead architect of the first PowerPC microprocessor, the 601. Previously, he

was one of the key designers of the branch unit of the original RISC System/6000 chip set and was also heavily involved in the development of the RISC Single Chip (RSC) Power microprocessor.

Moore received a BSEE degree from Rensselaer Polytechnic Institute and an MSEE from the University of Texas at Austin. He is a member of the Institute of Electrical and Electronics Engineers and the Computer Society.



Michael C. Becker, who works for Motorola's RISC Division, led the Motorola designers that were part of the 601 microprocessor development team. Currently, he is continuing work on future PowerPC products. Previously, he worked for Data General and Netstal Machinery, Switzerland.

Becker holds a BSEE and an MSEE from Texas A&M University.



Michael S. Allen worked with IBM to define the 601 bus interface. He is currently designing the bus interface unit for an advanced PowerPC-based microprocessor at Somerset. Prior to his work in Motorola's RISC Division, he developed integrated microprocessor products at

National Semiconductor.

Allen holds a BSEE from the University of Texas at Arlington. He is a member of the IEEE.

John S. Muhich is a senior engineer for the IBM Corporation. He is the co-lead architect of the 601 PowerPC microprocessor and the RISC Single Chip (RSC).



David P. Tuttle is an IBM senior engineering manager and the PowerPC 601 microprocessor project manager. Earlier, he was manager of the RS/6000 processor architecture. He also served as the lead logic designer of the RIOS-1 data cache unit and, later, the design manager responsible for the RIOS-1 data cache unit and storage control unit. Tuttle holds BSEE and MSEE degrees from the University of Louisville and an MBA from the University of Texas at Austin.

Direct questions concerning this article to Michael C. Becker, Motorola, Inc., 9737 Great Hills Trail, Austin, TX 78759; becker%ibmoto.com@aimgate.sps.mot.com.; or to Charles R. Moore, IBM Corp., 9737 Great Hills Trail, Austin, TX 78759; cmoore%ibmoto.com@aimgate.sps.mot.com.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 168

Medium 169

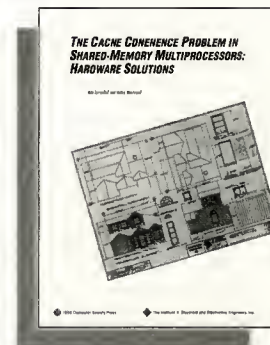
High 170

THE CACHE COHERENCE PROBLEM IN SHARED-MEMORY MULTIPROCESSORS: Hardware Solutions

edited by Milo Tomasevic and Veljko Milutinovic

This book provides insight into the nature of the cache coherence problem and the wide variety of proposed hardware solutions available today. It is a research and development reference for computer system architects, engineers, and designers. The first chapter introduces the shared-memory multiprocessor environment, details the cache coherence problem, and surveys existing solutions. Subsequent sections of the book

- ▼ Elaborate on the memory reference behavior of parallel programs
- ▼ Describe directory cache coherence schemes
- ▼ Explore the large range of snoop protocols for shared-bus multiprocessors
- ▼ Discuss aspects of cache coherence in multiple-level hierarchies
- ▼ Examine scalable schemes for large multiprocessor systems
- ▼ Evaluate different hardware coherence solutions
- ▼ Compare software solutions by means of simulation methods



Sections: Introductory Issues, Memory Reference Characteristics of Parallel Programs, Directory Cache Coherence Protocols, Snoopy Cache Coherence Protocols, Coherence in Multilevel Cache Hierarchies, Cache Coherence Schemes in Large-Scale Multiprocessors, Evaluation of Hardware Cache Coherence Schemes.

448 pages. July 1993. Hardcover. ISBN 0-8186-4092-8. Catalog # 4092-01 — \$62.00 Members \$50.00

To order call toll-free: 1-800-CS-BOOKS

FAX: (714) 821-4641 or E-Mail: cs.books@computer.org

IEEE COMPUTER SOCIETY PRESS



Spearmints:

Hardware Support for Performance Measurements in Distributed Systems

Spearmints permits fine-grained performance measurements in distributed systems with only minimal impact on the observed system. Each machine of the target system must have one sensor that collects relevant events and marks them with global time stamps. The sensors can be attached to a common measurement system that samples the marked events on- or offline, orders them chronologically, and analyzes the resulting sequence.

Uwe Kleinhans

Joerg Kaiser

Karol Czaja

*German National Research
Center for Computer
Science*

Distributed systems provide enhanced reliability, increased performance, and improved resource utilization. To take full advantage of these benefits and tailor a system to a given environment, users need detailed characteristics about the system's behavior. However, it is difficult to gain insight into the interaction and performance characteristics, because there is no global control of the activities in these systems.

Several projects used a method called event-triggered measurement to overcome this difficulty. Lange provides a detailed description of this method.¹ Generally, event-triggering provides detailed insight into a certain aspect of a system's behavior by analyzing a chronologically ordered sequence of logically triggered events. The time and place of each event must be known.

Researchers in the Incas project² at the University of Kaiserslautern, Germany, observe their system to detect performance problems as well as to support debugging in a distributed environment. To capture process-specific statistics (idle times, blocked times) as well as execution times for procedures and single statements, they analyze process-specific events (start, stop, block), procedure-specific ones (call, return), and executions of single statements.

Another example of an event-triggered measurement system is the Zaehlmonitor 4 monitor developed at the University of Erlangen-Nuernberg,

Germany.^{3,4} These researchers measure the duration of different phases in communication protocols and parallel algorithms by capturing the calls of and returns from the respective procedures as relevant events. To analyze the synchronization of tightly coupled processes and their communication, they observe single accesses to shared variables or to addresses used as semaphores for critical regions.

The Jewel measurement system focuses on an additional aspect in performance measurement that requires high-resolution observations.⁵ Jewel was developed in the Relax project at the Gesellschaft fuer Mathematik und Datenverarbeitung (GMD), the German National Research Center for Computer Science. It was used in the European Space Agency's Dosval project to examine different distributed operating systems for their usability in spacecraft.⁶ Jewel provides mean values about a system's performance in a number of events in a specific period of time. Furthermore, Jewel captures individual events to calculate empirical distribution functions, standard deviations, and variations for the analyzed events. Besides the overall performance, these numbers describe the reliability of a system's behavior—an important factor if the observed system is to be used under extreme conditions or in sensitive environments.

Low interference

In the field of system measurement it is crucial that the influence of the measurement on the

observed system's behavior be as low as possible. The degree of interference in the measured system, the so-called object system, should be very low so that neither the observation nor the running system itself is affected. Accordingly, pure software monitors are not suitable, since they perform the tasks of event-triggered measuring by executing appropriate code within the object system. For that reason, the projects we've described make the tasks of collecting, ordering, and analyzing events part of a measurement system that is separated from the object system. The code to evaluate and process events executes on standard hardware platforms separated from the object system. When an event occurs in the object system, the measurement system receives all the information necessary to identify and evaluate it. As a consequence, the interference in the object system is reduced to the overhead produced by signaling events to the measurement system.

Global time

A prerequisite for the chronological ordering of events in distributed measurement is the notion of global time. To derive performance characteristics, we must be able to calculate the duration between events independently of their places of occurrence. Therefore, each event's moment of occurrence must be fixed according to a global time reference. Obviously, the resolution and accuracy of this time base directly affect the granularity and accuracy that can be achieved for the measurements. Researchers in the mentioned projects show that measurements in distributed systems often imply the observation of intervals with a duration in the range of the execution time for single statements. As standard time measurement facilities do not provide this resolution even for local measurements, they have developed dedicated hardware clocks. For the Jewel system we developed a system of local clocks that are synchronized to provide a global accuracy of about 0.5 μ s.

Global measurement support

Technical solutions to the problems of low-interference event detection and global time cannot be developed independently from each other. The way in which events are transferred to the measurement system not only determines the impact on the object system but also seriously affects the granularity as well as the accuracy of time measurements. The subsystem that samples events and the respective time stamps must impose minimal overhead on the object system. It also must guarantee that a time stamp denotes the time of an event's occurrence in the object system as accurately as possible. Any inaccuracy in this subsystem contributes directly to the inaccuracy of the whole measurement.

To provide the items of low interference and high accuracy for various combinations of an object system and a measurement system, we developed a component we call Spearminths,

a sensor for performance measurements in distributed systems. Spearminths is a means of gathering descriptions of the events in an object system, while requiring only minimal overhead within the observed system itself. It integrates an efficient mechanism for sampling events and a global time reference that causes minimal interference to the object system while supporting high-resolution measurements. The Spearminths distributed sensor receives events, marks them with global time stamps, and buffers these event records for further processing. It can be easily adapted and interfaced to an object system, and, as it is not tailored to a certain measurement system, the buffered event descriptions can be read by whatever measurement system is used for event processing.

System description

In designing the Spearminths hardware-based sensor, we focused on the essential issue of minimizing the impact of measurement on a system under test. We considered how best to generate complete event records, implement Spearminths as a distributed component, connect it to a local CPU, and interface it to a measurement system.

Generating event records. The contents of the event records generated by Spearminths depend on the information that is collected and stored for sufficient characterization of an event. In most approaches to event-triggered system measurement, an event is characterized by the time and the location of its occurrence as well as by information about its type and actual measurement data.^{2,3,5-7} The amount of information that has to be transferred from the object system to the measurement system should be as small as possible to meet the crucial objective of low interference.

The measurement system itself can fetch the time and location of an event's occurrence because they are visible from the outside of the object system. As this approach causes no impact on the system under test, most of the existing measurement systems work this way. The location of any event signaled to the measurement system is the place it is signaled from, while the event's time is gathered from a clock in the interface component that is accessed to signal an event. As the time stamps are used to order and compare events, the clock must produce them according to a global time reference. (The box summarizes GTR basics.)

In general, the type of an event and the actual data of interest associated with it depend on the current state of the object system. The only way to gather this information without any effect on the object system is to use a pure hardware monitor for observing the physical connections in the system under test. This approach is based on the assumption that all relevant events and measurement data can be derived from signal patterns appearing on the physical connections (pattern-matching triggering). Upon recognizing a signal pattern that indicates an event, the hardware monitor stores the pattern and associated data as the description of the event, along

Basics for a global time reference in distributed systems

Generally, system time measurement involves the identification of discrete points of time on a time axis and calculation of the intervals between them. The time axis is unique in the system and starts at one distinguished starting point T_0 . Each point X of time on this axis is identified by the time $\mathcal{T}(X)$ that has passed since T_0 . We measure $\mathcal{T}(X)$ by summing the pulses generated from an oscillator with a nominal frequency f_{nom} in the interval from T_0 to X . With $n(X)$ being the count of pulses, $T'(X)$, the measured time at point X is given by $T'(X) := n(X) * f_{nom}^{-1}$.

This leads to a least digit inaccuracy f_{nom}^{-1} , which is ignored in the following, because it is inherent to all kinds of measurements. As every oscillator deviates by factor f from its f_{nom} , $\mathcal{T}(X)$ differs from $T'(X)$ by factor f . For every oscillator, a factor F that denotes the oscillator's maximal deviation f from its f_{nom} is given. Therefore, measured time $T'(X)$ is subject to an inaccuracy of

$$\begin{aligned} T'(X) * (1 - F) &\leq \mathcal{T}(X) \\ &:= n(X) f_{nom}^{-1} * (1 + f) \leq T'(X) * (1 + F) \end{aligned}$$

To measure the length $t(i)$ of an interval i , we calculate the time between its starting point S_i and its termination point T_i . Assuming an oscillator's deviation f to be stable, the measured values $T'(S_i)$ and $T'(T_i)$ both differ by the same factor f from their real values $\mathcal{T}(S_i)$ and $\mathcal{T}(T_i)$. Consequently, the measured length $t'(i)$ differs by the factor f from the interval's real length $t(i)$. This results in an inaccuracy $t'(i) * F$ for the measured length:

$$\begin{aligned} S_i: \quad \mathcal{T}(S_i) &:= n(S_i) * f_{nom}^{-1} * (1 + f) = T'(S_i) * (1 + f) \\ T_i: \quad \mathcal{T}(T_i) &:= n(T_i) * f_{nom}^{-1} * (1 + f) = T'(T_i) * (1 + f) \\ t'(i): \quad t'(i) &:= T'(T_i) - T'(S_i) \\ t(i): \quad t(i) &:= \mathcal{T}(T_i) - \mathcal{T}(S_i) = t'(i) * (1 + f) \rightarrow \\ &t'(i) * (1 - F) \leq t(i) \leq t'(i) * (1 + F) \end{aligned}$$

A global time reference that is to be used throughout a distributed system to measure time with an accuracy of A , must implement the following two items:

- T_0 is the unique starting point for time measurement

throughout the system.

- The duration $t'(i)$ measured for any interval i must not differ by more than A from the duration $t(i)$ elapsed in reality. This must hold independently from the sites where its starting point S_i and its termination point T_i are fixed.

An event's time stamp should denote its time of occurrence as accurately as possible. Using a central clock as a global time reference subjects the time for the transmission of the time stamp to the event's location to significant variations that directly contribute to the measurement's inaccuracy. The variations depend on the communication subsystem used and may even be unpredictable (for example, Ethernet). To avoid these variations, local clocks in the nodes of the distributed system must generate the time stamps.

Due to physical effects that restrict the transmission of high-frequency signals over longer distances, the local clocks cannot use a central oscillator. Therefore the problems of start delay and differing real frequency have to be tackled to implement a global time reference using a system of independently triggered local clocks.

Start delay. Because of the finite speed of communication, two clocks J and K cannot be started simultaneously, but they differ by a specific value $\Delta t_{start}(J, K)$. If T_i and S_i of some interval i are fixed with clocks J, K , the difference $T'(T_i) - T'(S_i)$ has to be corrected by $\Delta t_{start}(J, K)$ to get the length $t'(i)$: $t'(i) = T'(T_i) - T'(S_i) + \Delta t_{start}(J, K)$. With U_{start} being the maximum uncertainty in the knowledge of all values $\Delta t_{start}(J, K)$ throughout the system of local clocks, each measured duration $t'(i)$ is subject to an inaccuracy of $\pm U_{start}$.

Differing real frequency. As every oscillator deviates by a specific factor from its nominal frequency, the times $T'(T_i)$ and $T'(S_i)$ measured with clocks J, K differ by different factors f_j and f_k from the real values $\mathcal{T}(T_i)$ and $\mathcal{T}(S_i)$. The measurement of $t(i)$ follows:

$$\begin{aligned} t(i) &:= \mathcal{T}(T_i) - \mathcal{T}(S_i) \\ &= T'(T_i) * (1 + f_j) - T'(S_i) * (1 + f_k) \\ &= t'(i) + T'(T_i) * f_j - T'(S_i) * f_k \end{aligned}$$

continued on next page

with a time stamp and an eventual location identifier. This method allows the hardware monitor to transparently recognize an event in the object system.⁷

However, pure hardware monitoring is not suited for general measurements, particularly for those in higher system levels. Because virtual memory management and context switches are transparent to a hardware monitor, it has no

information about the context of a particular pattern observed. Thus, relying solely on the observation of physical signal patterns is inadequate for recognizing events that are relevant on higher system levels. Instead, the object system must provide the measurement system with all the information that is necessary to identify these events. For this reason the code of the object system must be equipped with I/O rou-

Basics (cont.)

As there is a given upper bound $\pm F_{\max}$ for all oscillators in the system, the oscillator's different deviations from the nominal frequency lead to a maximal inaccuracy $D_{\text{freq}}(i) = 2t'(i) * F_{\max}$ in measuring an interval i .

To meet the requirements for a global time reference with an accuracy of A , the system of local clocks must contain mechanisms that guarantee that the following condition is not violated for any pair of clocks and any interval i :

$$|t(i) - t'(i)| \leq 2t(i) * F_{\max} + U_{\text{start}} < A$$

The common start of all local clocks must guarantee the upper-bound U_{start} for the uncertainty, to which the delay in the common start of some pair (J, K) of clocks is known. U_{start} has to be small in relation to the required accuracy of A . Intending a resolution of less than 1 μs , as used in today's measurement systems, the common start must be implemented using a communication system with almost definite transmission delays (point-to-point connections, Token Ring).

Synchronization is necessary to enforce that for any interval i the deviation $D_{\text{freq}}(i)$ of any pair of clocks is limited to $D_{\text{freq}}(i) = 2 * t'(i) * F_{\max} < A - U_{\text{start}}$. We achieve this by synchronizing the clocks after the common start with a fixed frequency f_{syn} . After adjusting to the time $n * f_{\text{syn}}^{-1}$ on the last synchronization, we set every clock to the time $(n + 1) * f_{\text{syn}}^{-1}$ when the next synchronization occurs. Thus, each pair of clocks can deviate only in the interval between two consecutive synchronizations. This limits the clocks' maximum deviation to $D < 2 * F_{\max} * f_{\text{syn}}^{-1}$. Assuming a maximal frequency inaccuracy of F_{\max} , we synchronize the clocks with a frequency f_{syn} of $f_{\text{syn}} > 2 * F_{\max} / (A - U_{\text{start}})$ to achieve a resolution of A .

times at those points where relevant events are to be signaled (embedded-code triggering). Upon executing such a routine, the object system writes both an event's type and its associated measurement data to the measurement system. Spearminits receives this information and combines it with the event's time. The resulting event record comprises all data that is necessary to analyze the event and stores it for further processing by the measurement system.

Spearminits as a distributed component. According to the discussion in the Basics box, the global time reference must be implemented as a distributed component to minimize any inaccuracy of the events' time stamps. Furthermore, the overhead in the object system, which is necessary to transfer information about an event to Spearminits, determines the

degree of interference due to signaling an event. Providing only insufficient bandwidth or using an interface with long access times may result in a significant reduction of the object systems' performance. Consequently, Spearminits consists of multiple sensors, each of them interfaced to one node of the object system. Every sensor contains a local clock that is synchronized with the clocks of the other sensors to constitute a global time reference. If an event occurs on a node, the object system transfers the appropriate information to the local Spearminits sensor that generates the event record and stores it. The sensors connect to the measurement system that fetches the event records for further processing. As an event record comprises all information about an event, this transfer is no longer critical to the measurement's accuracy.

A Spearminits sensor and the object system. Current distributed systems generally place nodes on workstations. Their architectures provide two classes of interfaces that meet the conditions necessary for transferring information from a node of the object system to the local component of Spearminits. They are the CPU and the system bus interfaces.

Connecting Spearminits directly to the local CPU in the object system's nodes yields a minimal delay for signaling an event. If the CPU interface is not arbitrated for multiple masters, this delay only results from restrictions for transmission over the physical connections. As it is small in relation to the magnitude of the accuracy desired for the measurements, the transfer over the CPU interface does not impact the time stamp's accuracy.

To use simple store instructions in the object system to signal events, the local Spearminits sensor must be integrated into a local address space that is visible and can be written to in all modes of computation. Using today's RISC CPUs, events can be signaled using store instructions with an overhead of 1 cycle per byte. Experiments with an object system signaling events with a rate of about $5 * 10^{-4}$ events/cycle and an data rate of 8 bytes/event^{5,6} show that the measurements lead to a negligible overhead of about 0.5 percent for event signaling. As a result, connecting a component of Spearminits to the local CPU's interface contributes to a very high degree to the objectives of low interference and high resolution.

In most of the hardware platforms used today, it is quite easy to attach the sensor to the system bus such as an Sbus, Turbochannel, or VMEbus. As long as this connection is exclusively used by the CPU and the sensor can be mapped into a directly accessible address space, the system bus may be used as well as the CPU interface. Using an address space that can be accessed only by system calls or explicit I/O routines results in an additional overhead for calling and executing this additional code to signal events. This overhead adds to the impact on the object system resulting from the measurement. Furthermore the variation in the execution times of system calls or I/O routines adds to the inaccuracy of the events' time stamps.

If the system bus is arbitrated between different components of the system, there are periods when an event cannot be signaled instantaneously because the bus is reserved for exclusive use by another component. If an event occurs during such an interval, the delay in signaling the event depends on the strategy employed in the channel's arbitration. As the arbitration strategies are not preemptive in general, the time to signal an event to the local Spearmints sensor may be subject to large variations that contribute directly to the inaccuracy of the event's time stamp. As the bus is not arbitrated, during, for example, burst transfers, an event signaled in this interval may be delayed by up to the duration of one burst transfer. This limits its time stamp's accuracy to the length of a burst. Typical values for a burst transfer from the disk to memory are 800 μ s in a VMEbus workstation and 5 μ s in a more sophisticated DEC 5000/200 system. This maximal delay in signaling an event not only influences the time stamp's accuracy but also slows down the object system's activities, thus implying a higher degree of interference.

Consequently, to achieve maximum accuracy for the time stamps and minimal interference in the system under test, the Spearmints sensors should be connected as closely as possible to the CPUs in the object systems' nodes. The described effects of bus arbitration have to be considered if the system bus of a node is used to integrate a Spearmints sensor into an object system. Generally, every Spearmints sensor should be mapped into an address space that can be directly accessed by every code running on the local node of the object system.

Spearmints sensor and the measurement system. The transfer of event records from the Spearmints sensors to the measurement system is not critical to the accuracy of the measurements and the impact on an object system. Instead, the frequency of events occurring in the object system guides the design of the interface between a Spearmints sensor and a measurement system. Its bandwidth has to be large enough to transfer the event records from the sensor to the measurement system before the sensor is flooded with event records. Furthermore, the measurement system uses this interface to program the sensors. Generally, the sensors should have an interface with sufficient bandwidth for bidirectional data exchange with a large variety of measurement systems running on various hardware platforms.

Architecture

According to these concepts, we implemented Spearmints as a system of sensors that constitute an interface between an object system and a measurement system. Each Spearmints sensor is attached to a node of the object system and to the measurement system. Furthermore, the local clocks of the sensors are linked by a synchronization channel that is necessary for the reliable and correctly timed transmission of synchronization signals. This is a prerequisite for the synchronization scheme used to implement Spearmints' global time reference.

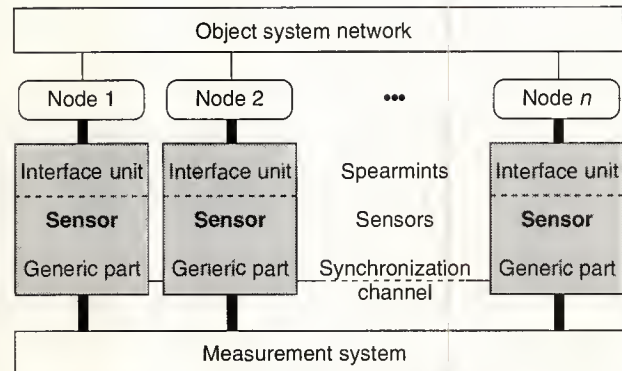


Figure 1. Spearmints' placement between an object system and a measurement system.

The overall architecture of Spearmints is shown in Figure 1.

A Spearmints sensor consists of an interface unit and a generic part. This separation provides the flexibility to integrate the sensor into different kinds of nodes. The interface unit depends on the architecture of the object system's node to which the sensor is attached. The generic part of a sensor executes the functions of creating and storing event records as well as passing them to a measurement system.

The different interface units mainly adopt standard bus interfaces and protocols. A sensor connected to these buses will be easily and quickly accessible from an object system, because they cover address spaces that can be mapped into the address spaces of user processes. If the object system writes to the interface unit of this sensor, the unit supplies the generic part with the address, the data, and a control signal that enables the sensor.

The generic part. Figure 2 (next page) depicts the structure of the sensor's generic part. This part is enabled by the sensor's interface unit when the object system signals an event by writing to an address that is assigned to the sensor.

As in many measurement systems,^{2,3,5,7} a part of the address that is accessed by the object system is interpreted as the signaled event's class. Thus, different event types can be distinguished. To support this feature, every Spearmints component occupies a range of consecutive addresses on the bus. Each access to this address space is interpreted as the signaling of an event. The address offset within this address space is interpreted as the event's type identifier.

Often an object system is observed under various aspects by doing multiple measurements, each of them focusing on different types of events. As it would be too cumbersome to change the instrumentation of an object system's code whenever a new aspect is to be observed, most systems that are subject to measurements are monitored continuously. That is, their code contains I/O routines for all kinds of events that

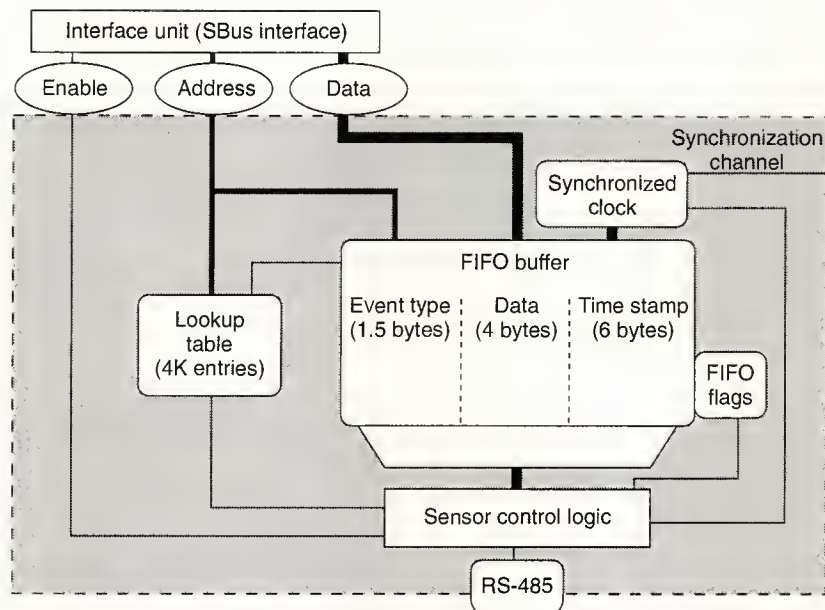


Figure 2. Structure of a sensor's generic part.

Table 1. The effect of our synchronization scheme.

Duration D (seconds)	Clocks not synchronized		Clocks synchronized	
	Deviation Δt	$F = \Delta t/D$	Deviation Δt	$F = \Delta t/D$
30	$2,803 \times 10^{-7}$	9.3×10^{-6}	0	0
300	$28,915 \times 10^{-7}$	9.6×10^{-6}	0	0
3,600	0.0344	9.5×10^{-6}	0	0
61,200 (17 hr.)	0.6	9.8×10^{-6}	10^{-7}	1.6×10^{-12}

may be of relevance for any measurement. The measurement system marks irrelevant event types in each Spearminits sensor's lookup table to prevent itself from being swamped with needless information. The sensor differentiates up to 4,096 types of events.

Upon receiving an event's type identifier, the generic part uses the lookup table to verify that it is a relevant event. If an event is relevant, the event's type identifier, the transmitted data word, and the actual count of the local clock are combined into one event record. The record is stored as one entry in the sensor's dual-ported FIFO buffer.

As Spearminits is intended to be used in today's workstations, we assumed a word length of 32 bits for the received data. If an event requires more than 4 bytes of data, additional accesses to the sensor become necessary. Dedication of a special event type for these accesses allows the resulting

entries in the FIFO buffer to be recognized as additional data for a previously signaled event. The FIFO buffer provides multiple flags that indicate the degree to which the buffer is filled. These flags can be polled by the measurement system that empties the buffer before an overflow can occur.

The local clocks of all Spearminits sensors constitute a global time reference with a resolution of 100 ns (see adjacent box). The clock's count width of 48 bits enables the generation of unique global time stamps for more than 7,800 hours. A bidirectional serial interface (RS 485) provides the communication channel between the sensor and the measurement system. With an assumed mean frequency of events of about $10^4/s^{2,5,6,8}$ and an average of 11.5 bytes of data associated with one event,⁵ the transfer of event records takes an approximate channel capacity of about 115 Kbytes/s. As bursts of events are buffered in the FIFO buffer, the serial interface's capacity of about 1 Mbyte/s meets this requirement. The sensor control logic controls the communication between the sensor and the measurement system via the serial interface. It interprets received messages and generates the appropriate control signals for the sensor's components.

Evaluation of the GTR. To evaluate and test Spearminits' global time reference, we have implemented a prototype consisting of a master-slave pair that is integrated into two DEC 3100 workstations. The clocks contain a 10-MHz oscillator, an 82C54 programmable counter, and two PAL22V10s containing the logic for synchronization and control. A simple fiber optics system serves as the synchronization channel.

To suffer as little access latency as possible, we attached the clocks to the workstations' EPROM sockets, which are directly connected to the CPU bus. We used the two clocks for measuring various intervals to verify our synchronization algorithm. We measured the duration D of each interval with the clocks running independently as well as being synchronized. For both cases we calculated the absolute deviation Δt of the clocks and the relative deviation $F = \Delta t/D$. The results are shown in Table 1.

Spearmints' global time reference

Each of Spearmints' sensors comes with a local clock that measures time by counting the pulses generated by the clock's oscillator running at frequency f_{nom} of 10 MHz. (See Figure A). To constitute a global time reference, the clocks exchange appropriate signals via a dedicated synchronization channel. The literature presents various methods for synchronizing clocks. They can be generally classified as either software synchronization based on the exchange and evaluation of messages or hardware synchronization using physical signals and effects.²⁻⁴

Software synchronization generally provides more flexibility regarding the clocks' interconnections. However, the software-based approaches presented up to now do not achieve a global resolution better than 100 μ s.¹ This is mainly due to the communication necessary for software synchronization being quite complex and time consuming. Furthermore, many communication schemes cannot even guarantee an upper bound U_{start} for the uncertainty in the delay of the common start and the synchronization (Ethernet).

To implement Spearmints' global time reference, we modified a hardware-based method mentioned in Zieher and Zitterbart,³ which is based on the idea of having one master clock and several slave clocks in the system. After the common start of all clocks, the master clock sends a synchronization signal with a frequency of f_{syn} , causing the slave clocks to synchronize themselves to the master.

As we wanted to trade the synchronization channel's reliability for its costs, we extended this synchronization scheme to tolerate the loss of a certain amount of synchronization signals. The measurement system executes the Reset instruction to prepare the master clock for global time measurement. The master clock disables its counter, resets it, and generates the RESET signal. Upon receiving this signal, the slave clocks lock and initialize their counters. Upon receiving a Start instruction, the master clock unlocks its counter and makes the slaves do the same by emitting the START signal. Once all clocks are started, the master clock generates the two synchronization signals SYNC-A and SYNC-B.

Additionally, we equipped the system of local clocks with a global stop mechanism. Executing a global stop instruction for the master clock lets this mechanism suspend further counting and generate a STOP signal that halts the slave clocks. Between a global start and a global stop, the clocks are synchronized ac-

cording to Spearmints' synchronization algorithm. This algorithm guarantees that no pair of clocks deviates by more than $2 * F_{max} * f_{syn}^{-1}$, if $F_{max} < 0.5$ (usually, relevant oscillators have an $F \leq 10^{-5}$):

- The master clock generates the synchronization signals alternatingly with frequency f_{syn} . The nominal duration $T_{syn} = f_{syn}^{-1}$ between two successive synchronizations is known throughout the system of clocks.
- At time $2n * T_{syn}$, with $n = 0, 1, 2, \dots$, the master clock generates signal SYNC-A. At time $(2n + 1) * T_{syn}$, with $n = 0, 1, 2, \dots$, it generates signal SYNC-B.
- Every time a slave clock receives a synchronization signal, its reaction depends on the received signal and its actual local time t_s (see Table A).
- As long as a slave clock does not receive a synchronization signal, it runs locally.

This algorithm guarantees that the clocks in the system resynchronize to a unique global time, if no more than $1/2 * F_{max}^{-1} - 1$ consecutive synchronization signals have been lost. With $t(mas)$ being the time at the master clock when synchronization comes up again, successful resynchronization takes place as follows:

continued on next page

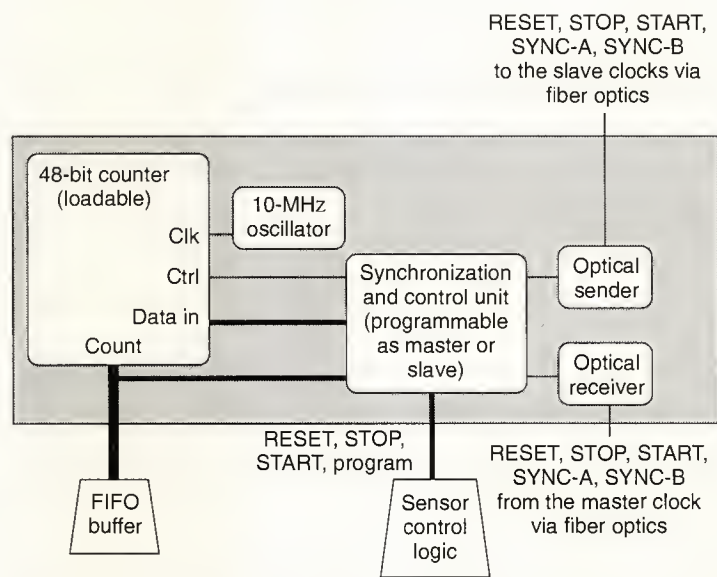


Figure A. Local clock of Spearmints' global time reference.

Spearmints' global time reference (cont.)

Table A. Signal reactions.

If signal is	and if $n \geq 0$ exists such that	then slave is	and t_s has to be adjusted to
SYNC-A	$0 < (2n - 1) * T_{syn} \leq t_s < 2n * T_{syn}$	Too slow	$t_s := 2n * T_{syn}$
SYNC-A	$2n * T_{syn} \leq t_s < (2n + 1) * T_{syn}$	Ok or too fast	$t_s := 2n * T_{syn}$
SYNC-B	$0 < (2n - 1) * T_{syn} \leq t_s < 2n * T_{syn}$	Ok or too fast	$t_s := (2n - 1) * T_{syn}$
SYNC-B	$2n * T_{syn} \leq t_s < (2n + 1) * T_{syn}$	Too slow	$t_s := (2n + 1) * T_{syn}$

- With $t(mas) = 2 * l * T_{syn}$ ($l=0,1,2, \dots$), the master generates signal SYNC-A. Every slave clock adjusts itself to the time $2 * l * T_{syn}$, if the following condition holds for its local time $t(slv)$: $(2 * l - 1) * T_{syn} < t(slv) < (2 * l + 1) * T_{syn}$.
- With $t(mas) = (2 * l + 1) * T_{syn}$ ($l=0,1,2, \dots$), the master generates signal SYNC-B. Every slave clock adjusts itself to the time $(2 * l + 1) * T_{syn}$, if the following condition holds for its local time $t(slv)$: $2 * l * T_{syn} < t(slv) < (2 * l + 2) * T_{syn}$.

In both cases the condition for correct resynchronization requires $|t(mas) - t(slv)| < T_{syn}$. As each pair of clocks deviates by a maximum rate of $2 * F_{max} * t$, this condition holds if the synchronization is not down for longer than $F_{max}^{-1} * T_{syn} / 2$. That is, up to $1/2 * F_{max}^{-1} - 1$ synchronization signals may be lost without affecting a successful synchronization.

Its synchronization and control logic controls each local clock. By setting a flag in the control logic, the clock can be programmed as the master clock or as a slave clock of the global time reference. Depending on this flag, the control logic generates control and synchronization signals, or it receives them and adjusts the clock's counter accordingly. The master clock sends the serially coded signals via fiber optics to the slave clocks. Including the variations in

transmission times and the different switch times of the clocks' logic, the uncertainty in the signals' delay does not exceed 30 ns. Assuming an inaccuracy of the oscillators' frequency of $F_{max} * 10^{-6}$, the clocks are synchronized with a frequency of $f_{syn} = 29$ Hz to achieve the 100-ns accuracy of the global time reference.

References

1. Flaviu Cristian, "A Probabilistic Approach to Distributed Clock Synchronization," *Proc. Ninth Conf. Distributed Computing Systems*, 1989, p. 288.
2. R. Hofman et al., "Integrating Monitoring and Modeling to a Performance Evaluation Methodology," *Entwurf und Betrieb verteilter Systeme*, Springer-Verlag, Berlin, 1990, pp. 122-149.
3. M. Zieher and M. Zitterbart, "NETMON—A Distributed Monitoring System," *Proc. EFOC/LAN, Sixth European Fibre Optic Communications and Local Area Networks Exposition*, IGI Europe, June/July 1988, pp. 452-457.
4. Hermann Kopetz, "Clock Synchronization in Distributed Real-Time Systems," *IEEE Trans. Computers*, Vol. C-36, No. 8, Aug. 1987, pp. 933-940.

The comparison of the results underlines the necessity for clock synchronization. Running the clocks with the synchronization lets us experience a constant difference of 200 ns or 300 ns in the values gathered from the two clocks. This results from a constant delay of 200 ns in the start of the clocks and the least significant digit error that is inherent to all kinds of digital measurement. The absolute deviation between two unsynchronized clocks is a linear function of the measured time. The relative deviation equals the nominal inaccuracy F_{max} of the used oscillators. This experiment shows that our synchronization algorithm is suitable to achieve a global accuracy in the range of a few

hundred nanoseconds.

Moreover, we tested the robustness of Spearmints' synchronization scheme against the loss of synchronization signals. After starting the two clocks globally, we suppressed the generation of synchronization signals for some intervals. We experienced a steadily increasing deviation of the clocks. If the synchronization came up again within five minutes, the two clocks are resynchronized on a common global time and run synchronized again. This value matches the theoretical limit $1/2 * F_{max}^{-1} * T_{sync}$ for the tolerable duration of failing synchronization very well. In our experiments, the deviation

in frequency and the time between two succeeding synchronizations had values of $F_{\max} = 10^{-5}$ and $T_{\text{sync}} = 65,536 * 10^{-7}$ s. Consequently, we could tolerate the loss of synchronization signals for a maximum duration of about 5.5 minutes.

SPEARMINTS IS A SYSTEM OF HARDWARE COMPONENTS that can be easily interfaced to the nodes of an instrumented distributed system for monitoring or evaluation using event-triggered measurements. The design of Spearmints follows the idea of providing a simple and universal tool that causes little interference and furnishes highly accurate measurements in distributed systems. As Spearmints only requires standard interfaces for its integration into an object system and its connection to a measurement system, it permits a wide range of measurement systems for the evaluation of a variety of distributed systems. This versatility distinguishes Spearmints from other similar approaches that mostly are tailored to a certain object or a certain measurement system.^{2,4,7,9}

Our actual work focuses on the implementation of the sensor and its prototypical integration in GMD's Jewel measurement system. As far as Spearmints' global time reference is concerned, we used experiences from a prototype integrated into two DEC 3100 workstations and one that is already used by the Relax measurement system¹⁰ to perform measurements in VMEbus-based systems. These measurements take advantage of 640-ns-resolution global time stamps. Actually, GMD develops an ASIC version of the global time reference for the Spearmints' prototype, which aims at an overall resolution of about 500 ns. To achieve this using a nominal clock frequency of 10 MHz and a synchronization frequency of $F_{\text{syn}} > 30$ Hz, the sensors must be integrated into the object system via an interface that guarantees a maximum uncertainty in access times of about 250 ns. GMD plans to integrate this new generation of Spearmints sensors into Turbochannel and Sbus workstations.

Moreover, we are interested in the question of how the redundancy in the system of synchronized local clocks and the ability to detect synchronization faults as well as to recover from them can be exploited to provide a fault-tolerant global time base. Making the sensor's clocks accessible to the object system may result in a fault-tolerant global time base of high accuracy that can be used in the distributed object system as well as in the measurement system. Furthermore, a fault-tolerant global time base of high resolution may be of great advantage for distributed real-time systems. How Spearmints' global time reference can be used in this context is part of the present work. ■

References

1. Frank Lange, "Leistungsmessung in verteilten Systemen—Konzepte und Werkzeuge zur Instrumentierung," (Performance Measurement in Distributed Systems—Concepts and Tools for Instrumentation), GMD Studien No. 179, May 1990.
2. Dieter Wybraniec and Dieter Haban, "Monitoring and Performance Measuring Distributed Systems Under Operation," *Proc. ACM SIGMETRICS Conf. Measurement and Modelling of Comp. Systems*, Association of Computing Machinery, New York, 1988, pp. 197–206.
3. R. Hofman et al., "Integrating Monitoring and Modeling to a Performance Evaluation Methodology," *Entwurf und Betrieb verteilter Systeme* (Development and Operation of Distributed Systems), Springer-Verlag, Berlin, 1990, pp. 122–149.
4. R. Hofman et al., "An Approach to Monitoring and Modeling of Multiprocessor and Multicomputer Systems," *Proc. Int'l Seminar on Performance of Distributed and Parallel Systems*, 1988, pp. 91–110.
5. F. Lange, R. Krueger, and M. Gergeleit, "JEWEL: Design and Implementation of a Distributed Measurement System" *IEEE Trans. Parallel and Distributed Systems*, July 1992.
6. F. Lange et al., "Distributed Systems Validation Method—Final Report," GMD, Studien No. 208, 1992.
7. A. Mink et al., "Multiprocessor Performance Measurement Instrumentation," *Computer*, Sept. 1990, pp. 63–75.
8. Cui-Qing Yang and Barton P. Miller, "Performance Measurement for Parallel and Distributed Programs: A Structured and Automatic Approach," *IEEE Trans. Software Engineering*, Vol. 15, No. 12, 1989, p. 288.
9. Flaviu Cristian, "A Probabilistic Approach to Distributed Clock Synchronization," *Proc. Ninth Conf. Distributed Computing Systems*, 1989.
10. Reinhold Krueger et al., "The Relax Concepts and Tools for Distributed Systems Evaluation," GMD Studien No. 168, Oct. 1989.



Uwe Kleinhans works in the field of antivirus systems at Norman Data Defense Systems AS in Drammen, Norway. This article reflects his work at the German National Research Center for Computer Science, Gesellschaft fuer Mathematik und Datenverarbeitung (GMD), in St. Augustin, Germany. His main research activities lay in the field of

IEEE COMPUTER SOCIETY PRESS

EUROMICRO '93

5th Workshop on Real-Time Systems

June 22-24, 1993 — Oulu, Finland

Euromicro '93 covers state-of-the-art research and development in real-time computing. The proceedings consists of 44 papers covering a wide range of issues, such as modeling, architecture, scheduling, operating systems, prototyping, fault tolerance, and artificial intelligence.

Sections: Modeling, Artificial Intelligence, Scheduling, Prototyping, Operating Systems, Object-Oriented Approaches, Control Systems, Petri Net Models, Timing Analysis, Fault Tolerance, Architecture, Reactive Intelligent Systems.

296 pages. June 1993. Softcover. ISBN 0-8186-4110-X.
Catalog # 4110-02 — \$70.00 Members \$35.00

EUROMICRO '93

Workshop on Parallel and Distributed Processing

January 27-29, 1993 — Canary Islands

Discusses current research and future major trends in the field of parallel and distributed processing. The workshop covers issues that include algorithms, communications, hardware techniques, parallel languages, and developmental environments.

Sections: Esprit; Massively Parallel Processing, Architecture; Logic Programming; Methodologies, Tools, and Environments; Algorithms; Analysis and Evaluation; Communications; Applications; Fault Tolerance in Parallel Systems; Simulation.

568 pages. Softcover. ISBN 0-8186-3610-6.
Catalog # 3610-02 — \$90.00 Members \$45.00

1st INTERNATIONAL SOFTWARE METRICS SYMPOSIUM

May 21-22, 1993 — Baltimore, MD

This proceedings contains 15 research papers on the latest theories and practices in the use of quantitative methods in software engineering. It discusses current results in measurement theory, model building and reuse, empirical validation of metrics, practical use of metrics, and experimental software engineering.

Sections: Case Studies, Code and Maintenance Metrics, Metrics Frameworks, Future Directions.

168 pages. May 1993. Softcover. ISBN 0-8186-3740-4.
Catalog # 3740-02 — \$50.00 Members \$25.00

To order call toll-free:

1-800-CS-BOOKS

FAX: (714) 821-4641

E-Mail: cs.books@computer.org



system architecture, where he analyzed the software-hardware interface for problems of fault tolerance, security, and system measurement in distributed and object-oriented systems. To show how conventional hardware can be enhanced with regard to dedicated requirements resulting from this analysis, he developed different hardware components and implemented them prototypically using ASIC technology.

Kleinhans studied computer science at the University of Bonn, where he received his diploma.



Joerg Kaiser heads the CREW (Cooperative Engineering Workbench) project at GMD in St. Augustin; CREW concerns system support for cooperative group work in design and engineering applications. His research work involves computer architecture and operating systems with an emphasis on object orientation and fault tolerance. He has published over 20 papers in these areas, investigated fault-tolerant VLSI structures, and held responsibility for the design and implementation of hardware support for an object-oriented distributed computer system. He has also worked in the area of high-precision and low-interference system monitoring and measurement.

Kaiser received his diploma in computer science from Bonn University.



Karol Czaja works as an MS systems adviser at Pfeifer and Langen, Cologne, Germany. At the time this article was written, he was a researcher at GMD where he developed an access control monitor providing protection in persistent object-oriented systems. His main research interests include computer architecture and operating systems with an emphasis on object orientation and fault tolerance.

Czaja received his diploma in computer science from the University of Bonn.

Direct questions concerning this article to Uwe Kleinhans, Kristian Bogneruds vei 8, N-0956 Oslo, Norway; or mettek@novell.oih.no.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 165

Medium 166

High 167



How Does Processor MHz Relate to End-User Performance?

Part 2: Memory Subsystem and Instruction Set

We conclude our study of end-user performance begun last issue by describing the performance implications of the memory subsystem and the effect of instruction sets on path length. Measured performances on many systems support our initial claim that cycle time is not sufficient to determine performance.

Steven W. White

Phil D. Hester

Jack W. Kemp

IBM, Advanced Workstations
and Systems Division

G. Jeanette McWilliams

Performance InDeed!

In Part 1 we studied end-user performance by selecting two micro-processors that used distinctly different design philosophies to achieve comparable performance levels. The 62.5-MHz IBM RISC System/6000 Model 580¹⁻³ (called RS1 here) exemplified the instruction-level parallelism approach, while the 160-MHz DEC Alpha⁴⁻⁸ illustrated the aggressive clock rate approach. We defined the performance components as clock rate, cycles per instruction (CPI), and path length. After discussing clock rate, we examined the differences within and across functional units and how they affected CPI. Now, we describe the performance implications of the memory subsystems and the instruction sets' effect on path length.

CPI, memory subsystem effects

Part 1 assumed that all instructions and data reside in the first-level (L1) caches. It also assumed that the virtual address translation information required to access instructions and data resides in the translation look-aside buffers (TLBs). Pipeline effects and interactions were lumped into the infinite cache element of CPI. In addition to pipeline effects, cache and TLB miss penalties can contribute significantly to measurable or finite cache CPI.

We define our end-user performance CPI component as

$$\text{Finite cache CPI} = \text{infinite cache CPI} \\ + \text{finite cache effect} + \text{finite TLB effect}$$

The finite cache effect is the product of the average number of cache misses per instruction and the average cache miss penalty in cycles. The finite TLB effect is the product of the average number of TLB misses per instruction and the average TLB miss penalty in cycles. We further partition the finite cache effect between penalties for cache misses and the eventual updating of main memory for stores. The cache and TLB misses include both instruction and data references.

The unit for each of these quantities is cycles per instruction. If each term is multiplied by instruction count, our equation simply states that total measurable execution time is the sum of the pipeline cycles, the cache miss penalty cycles, and the TLB miss penalty cycles. Improvements obtained by reducing either the pipeline CPI or finite cache/TLB effects are amortized across the total execution time.

L1 cache miss counts. In attempting to minimize the number of cache misses, a cache designer may make four main choices: set associativity, line size, capacity, and store policy.

Table 1. Simulated miss counts as a function of instruction and data cache capacity and geometry.

SPECint89 benchmark	Instruction cache		Data cache	
	Alpha (8 Kbytes, direct, 32-byte line)	RS1 (32 Kbytes, two-way, 64-byte line)	Alpha (8 Kbytes, direct, 32-byte line)	RS1 (64 Kbytes, four-way, 128-byte line)
Gcc	1,361,171	204,268	646,776	32,557
Li	227,991	819	515,430	1,557
Eqntott	86	51	344,742	146,030
Espresso	56,831	1,090	446,409	3,825
Total cache misses	1,646,079 @ 6 cycles	206,228 @ 10 cycles	1,953,357 @ 6 cycles	183,969 @ 10 cycles
Finite cache miss effect/instruction	0.082	0.017	0.098	0.015

High clock rates are more easily achieved by simplifying some aspects of the cache design. Options include smaller caches, direct-mapped caches, and store-through caches (and prohibiting byte and 2-byte accesses as well as unaligned accesses).

To see how clock rate goals affect these choices, consider set associativity. In an N -way set-associative cache, designers must compare appropriate address tags to determine which entry contains the requested data prior to its use. In direct-mapped caches, a common strategy is to use the only piece of data in the selected congruence class and start it down the pipeline. Then the processor cancels the operation if it is later determined that a cache miss occurred and incorrect data was being used.

In high cache hit environments the cancellation penalty is rarely paid, and this practice allows the designer a head start on getting the data to the functional unit, effectively shortening the pipeline. The drawback is that direct-mapped caches have higher miss rates than the equivalent set-associative caches, with the difference in miss counts being more pronounced for small (4-Kbyte, 8-Kbyte) caches.⁹ Therefore, designers often choose direct-mapped caches when the cache is large (and associativity is not as important) or when access time is critical.

The second key design lever for reducing cache misses is cache capacity. Capacity is often limited by cycle time goals since cache and TLB accesses are often on several critical timing paths. As cycle time is decreased, designers must either add cycles per cache access or reduce the time (in nanoseconds) to access the cache. The former results in more load-use delay as illustrated by the two-cycle delay for the 62.5-MHz RS1 when compared to the three-cycle delay for the 160-MHz Alpha.

Physical and electrical constraints, which worsen as cache capacity increases, limit the access time. We can reduce the

physical access time for the cache by decreasing the capacity of the cache. Physical access time can also be reduced by using only on-chip cache to avoid chip crossings; however, this also places a limit on capacity. (The assumed inverse relationship between clock rate and L1 cache capacity is supported by comparing the 32-Kbyte instruction cache (I-cache) and 64-Kbyte data cache (D-cache) on the 62.5-MHz RS1 to the 8-Kbyte instruction and data caches on the 160-MHz Alpha.)

For a given main memory (or L2) with a fixed access time in nanoseconds, increasing the processor clock rate has a compounding effect on cache. It forces designers to smaller caches that have higher miss rates, and it increases the number of processor clocks required to satisfy a miss.

Table 1 illustrates the effects of set associativity, line size, and capacity for the four SPECint89¹⁰ codes. (These effects are based on RS1 access patterns and vary with compiler.) The table lists the results of cache simulations for snapshots of the SPECint89 work loads. Our simulations used three trace snapshots (10 million instructions each) from each of the four benchmarks. Traces of this length are adequate to render the cold cache start-up effects negligible; for an 8-Kbyte cache with 32-byte lines, there can be at most 256 additional misses due to cold cache effects. The additional miss penalty is three or four orders of magnitude smaller than the execution time of the trace. (The store instructions were filtered from the traces prior to modeling the Alpha D-cache to account for Alpha implementing an allocate-on-read cache.)

The simulated miss counts reflect RS1 code. Some aspects of the Alpha design have potential for further reducing the effectiveness of the 8-Kbyte I-cache. To fill the load, branch, and functional unit delays, the compiler writer must perform aggressive loop unrolling and in-lining.⁶ This increases the footprint or instruction working set (not the dynamic path

length) of the application and can cause more I-cache misses than would occur with tightly rolled loops and subroutine calls. In processors with small caches, the choices may be unattractive: to stall for the pipeline delays or to wait for an I-cache miss to be satisfied. Based on our experiences in moving from the 8-Kbyte I-cache on the original RISC System/6000s (1990) to the 32-Kbyte I-cache on the current RS1, opportunities to gain performance by code expansion are much more limited in the smaller I-caches.

L1 miss penalties. The average L1 cache miss penalty is often the part of the finite cache effect that is most difficult to determine. Due to interaction between the pipelines and caches, not all misses incur equal penalty. If instruction prefetch triggers the miss early enough, and if instructions already fetched can provide work for the functional units while the miss is being satisfied, an I-cache miss is completely hidden. Hidden misses do not result in a penalty.

At the other extreme, a data dependency may cause the pipelines to stall the cycle after a cache miss, resulting in the maximum penalty. Additional functional units may decrease the infinite cache CPI and allow the dispatched work to be completed earlier. However when a cache miss is outstanding, finishing the work earlier may simply mean stalling the pipelines for more cycles. Pipeline and cache effects are not independent; cache and TLB misses can partially or completely negate the benefit of additional units.

Penalties for the cache misses shown in Table 1 can be subdivided into leading-edge and trailing-edge categories. The leading-edge portion includes cycles that elapse between the start of the miss and the first transfer. In both designs, this first transfer contains the requested data (also known as critical word first). The leading-edge time is dependent on main memory (or L2) access time. The trailing-edge includes the remaining cycles of the miss, to transfer the rest of the cache line. Trailing edge is a function of the cache line length, the width of the cache-memory bus, and the number of processor cycles per transfer.

Both designs include a number of features to reduce the penalty of I-cache and D-cache misses. A store-back buffer in RS1 reduces the leading-edge penalty for D-cache misses. When a D-cache miss causes a dirty (or modified) line in the cache to be selected for replacement, main memory must be updated with the modified data. Without a buffer, this memory update, or cast-out operation, must be made prior to the cache miss request so that room in the cache is available for the returning miss data. In RS1, the modified cache line data is placed in the store-back buffer, allowing the cache miss request to be started immediately. The bus and memory transaction for the dirty line is delayed until the bus is idle or the next line is cast out.

RS1 reduces the leading-edge effects of an I-cache miss with branch look-ahead logic (discussed in Part 1). Additionally, the fixed-/floating-point unit queues allow the

Performance metrics

When deciding how to judge the performance of a processor design, remember to select work loads that are similar to those of the intended user of the design. We selected the SPEC suite of benchmarks because they include a variety of complete applications from the workstation/server user community. SPEC closely controls source code and mandates full disclosure of variances and configurations. This approach provides a fair basis for system comparisons.

In our study we used metrics based on two SPEC suites. SPECmark89 is the geometric mean of performance on 10 benchmarks, four fixed point and six floating point. In 1992, SPEC released a second suite that carries separate fixed- and floating-point metrics: SPECint92 and SPECfp92. These metrics are geometric means of performance on six integer benchmarks and 14 floating-point benchmarks. With the exception of a matrix multiply program, all members of the original suite are included in the newer one. The original benchmarks were improved for the second suite (portability changes, new input files to provide runtimes long enough to be easily repeatable on newer faster machines, and so on).

Vendors of most systems in the server/workstation market have reported performance on both suites, so we have a good body of reliable data on many systems.

If we were just now starting the work that led to this article, we would rely entirely on SPECint92, SPECfp92, and Linpack to characterize performance. (We would include Linpack because it represents a large body of engineering/scientific applications that the matrix code covered in the old suite.) When we made simulations to support some of our observations, we only had traces on the older suite.

When we present actual performance results for the two architectures under consideration, we show all four metrics (SPECint92, SPECfp92, SPECmark89, and Linpack).

branch unit to proceed ahead of the arithmetic units. As a result, some of an I-cache miss leading-edge penalty can be overlapped with previously dispatched work.

Alpha saves some D-cache leading-edge cycles with a limited form of hit-under-miss logic. Hit under miss allows subsequent storage reference instructions to access the cache before the first transfer of the missing line arrives. An Alpha D-cache miss condition is recognized when the load reaches stage 6 in the load and store pipeline. (Stage 3 is dispatch.) When a D-cache miss occurs, Alpha blocks issue for a class of instructions that includes loads and stores. The only can-

***Due to interaction between
the pipelines and caches,
not all cache misses incur
equal penalty.***

didates for hit-under-miss processing are the two storage references—if present—in stages 4 and 5 (already issued), if they are D-cache hits. If either is a miss, it is queued in a “silo.” All subsequent storage reference instructions are blocked from dispatching until the silo empties and the first transfer from the final miss arrives. This strategy eliminates up to two cycles of the leading-edge penalty.

To hide some leading-edge effect on I-cache misses, Alpha includes an I-cache stream buffer used only for the sequential execution path.⁴ When an I-cache miss occurs and the line is not in the stream buffer, Alpha generates a memory (or L2) request. If the line is in the stream buffer, the miss is satisfied sooner. In either case, Alpha generates a prefetch request for the next sequential line. Although this may reduce the leading-edge effects in sequential code blocks, it may increase memory (or L2) use and bus contention with extraneous requests in branch-rich codes such as gcc. (It is not clear to us whether stream buffer requests to the L2 can be canceled if it is determined that a branch has been taken.)

Alpha's trailing-edge penalty may be very small. In one mode, 16-byte transfers per bus cycle (at least two processor cycles) can fill a 32-byte line in four (or more) processor cycles. (Alpha supports bus frequencies between one eighth to one half of the on-chip processor clock.) As 16-byte transfers arrive from the bus, they are moved, 8 bytes per processor cycle, into a “pending fill” latch. During this time, the cache remains accessible to other requests. When the entire line has been accumulated in the pending fill latch, it is dumped into the cache. The cited references do not state whether data (other than the requested operand) is accessible from the pending fill latch before it is transferred to the cache. It is also not clear whether the I-cache or a returning I-cache line is accessible during an I-cache line fill.

RS1 benefits (in the area of cache hits) from long D-cache lines at the expense of a moderate reloading period of eight cycles. To reduce the D-cache trailing-edge penalty, RS1 supports hit under fill with a cache reload buffer (CRB), which is similar to the Alpha pending fill latch. Hit under fill allows subsequent accesses to the cache once the first word returns on a cache miss. Data is accessible from the CRB as it returns. RS1 hides some of the I-cache trailing-edge effect by for-

warding the returning instructions to the fetch pipeline as the I-cache fills.

For the SPECint89 traces, simulations provide the I-cache and D-cache miss counts previously shown in Table 1. The performance impact of these misses can be roughly estimated by assuming a miss penalty. This miss penalty should take into account the just-mentioned features, which aim to reduce the leading-edge and trailing-edge effects. It should also consider the frequent data dependencies in the sampled application. Note that in both RS1 and Alpha, a functional unit pipeline stalls when an instruction requires the operand being returned on a cache miss.

Assuming the L2 cache satisfies all of the L1 misses, the Alpha L1 cache miss penalty is estimated at six processor cycles, 38 nsec at 160 MHz. At the fastest designed bus interface—half the processor clock—this is three 80-MHz Alpha bus cycles: request onto bus, L2 cache access, and data on bus.

The limited hit under miss may shorten the cache miss penalty by two cycles. However, the cache penalty may be lengthened by many factors such as data dependency, L2 and bus contention, and back-to-back misses. (Only the first miss obtains the hit-under-miss benefits.) For comparing the performance effects of the RS1 and Alpha caches, we simply chose six processor cycles as the cost for an Alpha cache miss. A pure comparison of cache designs would use the same six-cycle miss penalty for the RS1 design point. However, our exercise attempts to compare the effects of processor clock on performance and therefore includes effects of an L2 (in Alpha) to keep memory relatively close as the processor frequency is increased. For the moderate clock rate RS1, we used 10 processor (or RS1 bus) cycles (160 nsec) for L1 misses (satisfied by main memory). For each type of cache, the last row in Table 1 shows the estimated finite cache miss effect. We calculated it as total penalty cycles (total misses times miss penalty) divided by total instruction count.

L1 store-back penalties. In addition to the finite cache miss effect, the finite cache effect includes the penalties associated with writing stores to main memory. Alpha is somewhat unique among high-end RISC microprocessors in that it uses a store-through policy for the L1 D-cache; a four-entry (32 bytes per entry) write buffer reduces the store bus traffic.⁴ Rather than delaying the effects of a store by modifying a cache line and then casting out changed lines as they are selected for replacement, stores modify the contents of the write buffer entries. At a later time, the processor transfers write buffer data to memory. Since the write buffer entries are the same length as an Alpha cache line, the four-entry write buffer provides a function similar to four RS1 store-back buffers.

Alpha does not support unaligned stores or stores of less than 4 bytes, so each of the 32-byte entries requires only 8 valid (and therefore dirty) bits. Each valid bit covers a cor-

responding 4-byte portion of an entry. When a buffer entry is emptied, the valid bits appear to the memory (or L2) as mask bits, which control the choice of 4-byte quantities to be written. Therefore, the original cache line data need not be fetched on a store miss. Alpha implements an allocate-on-read miss policy.

Until an entry transfers to memory, additional stores to the same cache line simply update the appropriate write-buffer entry. Head and tail pointers maintain a rough FIFO relationship among the four entries. The write buffer attempts to write its head entry to memory whenever the write buffer contains at least two valid entries—not when the buffer is full.

Since the packaging of stores in the Alpha write buffer is not deterministic, a cycle-by-cycle Alpha simulation including caches would be required to accurately determine their effectiveness. We did not model this. We modeled four 32-byte write buffer entries assuming a least recently used (LRU) replacement algorithm. (This should underestimate the Alpha store traffic since the model does not purge an entry until the four-entry buffer overflows. Alpha, however, might purge an entry when there are only two valid entries.)

We counted the number of stores and the number of times the LRU buffer entry had to be emptied to make room for a new store request. Our previously mentioned cache simulations also provide cast-out counts, assuming store-in designs. Table 2 compares these results for the same SPECint89 traces we've described. The number of stores represents the store-through bus traffic without the write buffer. The effectiveness of the write buffer in minimizing store bus traffic is excellent (1/600) in the eqntott benchmark, fair (1/6) in espresso, and only marginal (1/2) for gcc and li. The write buffer reduces total store bus traffic for these samples (120×10^6 total instructions) by a factor of 2.5.

Alpha's small store-through cache and write buffer have significantly more store bus traffic than RS1's relatively large store-in cache. Determining the end-user performance impact is difficult since the amount of penalty per bus transaction depends on the likelihood that the bus (or memory) is busy when the transaction is requested, as well as the resulting waiting period. Therefore, applications that have larger miss rates and the associated higher bus/memory use will experience larger cast-out penalties. The two main effects of the write buffer (compared to store-in) approach are the additional latency incurred by some stores (to a full write buffer) and the additional use of the memory subsystem. Increased use of the memory subsystem will increase

Table 2. Write-back bus traffic.

SPECint89 benchmark	No. of stores	Four 32-byte LRU write buffers	RS1 (store-in, 64 Kbytes, four-way, 128-byte line)
Gcc	3,322,758	1,390,366	12,176
Li	4,539,836	2,145,440	1,283
Eqntott	369,647	632	164
Espresso	1,318,309	224,940	3,064
Total lines written back	9,550,550	3,761,378 @ 4 cycles	16,687 @10 cycles
Write-back penalty cycles (per instruction)		0.125	0.001

the average penalty for cache misses.

For purposes of illustration, assume that a cast-out from the Alpha write buffer takes two bus (four processor) cycles. (The increase in L1 cache miss penalty, which results from the store traffic's effect on L2 use, is included in this cast-out penalty.) Table 2 shows a total of 3.76×10^6 cast-outs for Alpha; at four cycles each, this means 15×10^6 processor cycles.

For the RS1 design, the cast-out penalty is negligible when bus traffic is light, due to the store-back buffer. Although the miss and store activity is very light for the RS1 cases, for illustration we use a pessimistic value of 10 cycles per cast-out. The resulting RS1 penalty is two orders of magnitude smaller than the Alpha penalty estimate.

The finite cache effect for the four sets of SPECint89 samples is the sum of the finite cache miss effect (Table 1) and the store traffic penalty (write-back penalty cycles, Table 2). The Alpha finite cache effect would be 0.305 while the RS1 value would be 0.033—about a 9:1 ratio. The Alpha miss penalty estimate assumes the L1 miss being satisfied in six cycles; for systems with different L1 miss penalties, the total penalty cycles are approximately proportional. Two 200-MHz DEC Alpha systems with different off-chip memory subsystems provide an example of performance variance with an L1 miss penalty. The better memory subsystem gains about 5 percent in performance on the SPECint92 benchmark and 18 percent on SPECfp92.¹¹

L2 caches. Alpha's aggressive clock rate requires a good second-level cache to achieve the assumed L1 miss penalty of six processor cycles. We assume that all L1 misses are satisfied by the L2 in Alpha (infinite L2 cache) and main memory in RS1. However, the introduction of an L2 adds the possibility of L2 misses. L2 misses are relatively expensive; for an aggressive clock rate design, the main memory is a significant number (20 to 30) of processor cycles away. Since the pipelines are likely to stall during an L2 miss, we can

***The compound FMA instruction,
along with the update forms of
storage reference instructions,
gives RS1 a clear path length
advantage on floating-point
applications.***

estimate L2 miss effects by adding the total L2 miss penalty (cycles per L2 miss times the number of L2 misses) to the infinite L2 cache cycle count previously estimated. However, we make no estimate of the L2 miss counts or penalties.

To reduce the number of L2 misses, Alpha supports a prefetch into L2 instruction. Assume prefetch instructions are scheduled sufficiently far ahead of an L1 miss and sufficient memory-to-L2 bandwidth exists to sustain the transfers. Then, prefetching the "to be missed" data allows the L1 miss to be satisfied by the L2 without experiencing the additional penalty of an L2 miss. However, the performance impact of L2 misses may be significant in some applications. For example, large databases can be accessed in a semirandom (nonsequential) fashion. Since the databases are often orders of magnitude larger than affordable L2 capacities (or even main memory), the first access of each group of records is likely to result in an L2 miss. The randomness of the access patterns makes prefetching difficult.

A less obvious cause of L2 misses is a context switch. When many unrelated processes run in the interval between time slices for a given process, they may purge private data of that process from L1 and L2. Each time the process starts to run, it effectively has cold L1 and L2 caches. The L1 being purged is not as significant since the number of compulsory misses, and the resulting additional penalty, is small. However, due to its size, the number of cycles required to warm up the L2 may become a considerable portion of each time slice. (For a 1-Mbyte L2 with 64-byte lines and a 20-cycle miss penalty, the maximum additional penalty per time slice is 320,000 cycles. With time slice intervals on the order of a million cycles, this could be a significant overhead.) Therefore, a common assumption, that a data structure is established in L2 and the effects of L2 misses can be ignored, may not be valid under these conditions. In this particular situation, the use of the Alpha prefetch instruction to warm up the L2 is limited since neither the programmer nor a compiler could be expected to know when the time slice interrupts

occur during any specific run of the program.

TLB effects. In addition to infinite cache CPI and finite cache effects, TLB misses contribute to the finite cache CPI.

RS1 has a 128-entry, two-way set-associative instruction TLB and a 128-entry, two-way set-associative data TLB. Each RS1 TLB entry maps one 4-Kbyte virtual page to the corresponding 4-Kbyte real page. RS1 hardware resolves TLB misses in roughly 40 to 50 cycles. Alpha partitions instruction TLBs into two types: eight fully-associative, 8-Kbyte-page entries and four fully associative, 4-Mbyte-page entries. The Alpha data TLB consists of 32 fully associative entries, each mapping an 8-Kbyte, 64-Kbyte, 256-Kbyte, or 4-Mbyte page. Alpha software resolves TLB misses.

Hardware TLB reloadings, large TLBs, and a compiler's data access pattern restructuring reduce the RS1 data TLB miss penalty on the SPECmark89 benchmark to negligible levels. The granularity supported for the Alpha data TLBs, and assumed exploitation by the operating system, provide adequate tools to make Alpha data TLB misses negligible on SPECmark89.

Instruction TLB misses are not negligible. We don't know many details of Alpha's four 4-Mbyte instruction TLB entries; they may be reserved for operating system use.⁵ Since use of any of these entries requires blocking 4 Mbytes of contiguous real memory, and all 4 Mbytes must be paged in for the TLB entry to be valid, it does not seem likely that they would be allocated to a user task. If limited to only eight Alpha 8-Kbyte instruction TLB entries, significant miss rates can be expected on some applications. Using the instruction traces mentioned earlier, and an estimate of 100 cycles for a software TLB reload, simulations indicate about 25-30 percent degradation on the gcc benchmark. We may have understated this effect as the model assumes the eight instruction TLB entries are managed on an LRU basis. Alpha instruction TLBs are managed on a not-last-used basis.⁴ Using the 50-cycle RS1 TLB miss penalty and the larger instruction TLB, simulations indicate 4 to 6 percent degradation for RS1 on gcc.

Overall CPI. Combining the pipeline effects from Part 1 with these cache and TLB effects, we conclude the following about CPI for the SPEC89 suite. The finite cache CPI should be significantly higher for Alpha than for RS1. On fixed-point codes, although the pipeline CPI appears to be similar for RS1 and Alpha, the differences in cache effects on these codes are significant. Simulations show that Alpha, even with a fast L2 cache to satisfy L1 misses, has significantly more total miss penalty due to the smaller, direct-mapped L1 caches. If no L2 is present, the total Alpha miss penalty will be significantly higher than our estimates. On floating-point codes, we expect the pipeline CPI to be significantly higher for Alpha than for RS1. Alpha's smaller, direct-mapped, store-through caches (with short lines) will further widen the gap in finite cache CPI values for the SPECfp89 benchmark.

Figure 1 shows how CPI scales with the clock rate. The lower curve illustrates constant path length and CPI—idealized linear scaling of performance with frequency. As clock rate increases to the limit bearable by the pipeline stages, designers can achieve higher clock rates by decreasing the amount of work per stage. The next curve shows CPI increasing with the clock rate; the CPI increases result from effects such as increased interlocks between functional units and longer latencies that result from lengthened pipelines. When comparing this pair of processors, note that the effects of lengthening the pipeline are more pronounced in floating-point codes than in fixed-point applications. The upper pair of curves reflect added CPI effects of the memory subsystems.

Path length

We have presented the relationship between frequency and CPI in the context of a common instruction sequence for both RS1 and Alpha. Next, we examine the performance implications of some differences between the RS1 (POWER) and Alpha instruction sets. Instruction set architects consider hardware/software interaction when deciding how to partition functions between software and hardware. The instruction set itself may be influenced by the technology and cycle time expectations of the first implementation.

A subtle distinction exists between the clock rate and CPI differences previously discussed and the path length issues we discuss now. The former are implementation issues; the latter are architectural issues. Generally, designers determine implementation details, such as multiple functional units, pipeline lengths, and cache parameters, by the practical concerns of cost and speed. As technology changes, so do these features—even for a given architecture. Architectural issues, such as storage reference restrictions, special-purpose registers, and update forms of storage references, are much more difficult to alter as the software base grows. While the implementation details described earlier can change with each release of hardware, architecture tends to be more stable.

Path length is the number of instructions required to perform some function; it depends on the instruction set, the number of registers, and the compiler. Both machines have roughly the same number of registers. Since we are in no position to compare the compilers, we address the way the instruction set affects path length. In all cases, the function provided by one instruction in one architecture requires more than one instruction in the other. Obviously, the change in overall path length is a function of changes for specific sequences and their relative frequency in an application. Increased instruction path length does not proportionally decrease performance when the newly added instructions affect overall CPI.

Alpha's main path length advantage involves branches. Alpha branch instructions include some simple testing (odd, even, or compare to zero) of a general-purpose or floating-

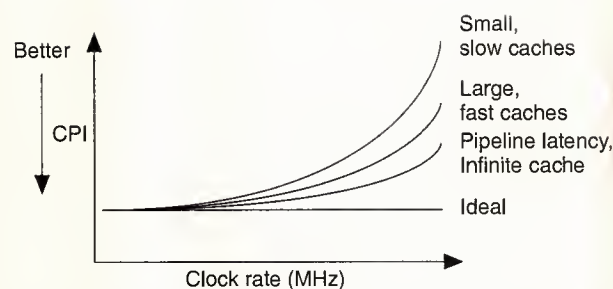


Figure 1. Effects of clock rate on CPI, including pipeline and memory subsystem.

point register. Determining one of these attributes for a value in a register does not require an explicit compare instruction. RS1 requires a compare and a branch to replace Alpha's branch in such a case. The impact of the additional compare is highly dependent on the ability to schedule. If any other attribute must be determined (for example, being equal to the contents of another register or to an immediate value), Alpha also requires an explicit compare instruction. Alpha's test and branch capability provides no path length advantage in such cases.

Alpha also provides a pair of conditional move instructions, which allow a simple register test (odd, even, or compare to zero) to determine whether a specified register copy operation is performed. In the worst case, RS1 uses a three-instruction sequence (compare, conditional branch around copy, copy) to replace this one instruction. Again, when the user needs to compare two values to determine whether the copy is performed, Alpha also requires a compare. In this case, the benefit of a conditional move is the elimination of a forward conditional branch. The conditional move cannot be used to replace loop-closing branches, subroutine calls, and goto constructs. For some of the remaining RS1 branches, the associated compare is scheduled so that the branch is resolved without guessing. Some unresolved branches are predicted correctly, incurring no penalty; the remaining branches incur a one- to three-cycle penalty. By removing one, two, or three penalty cycles on some branches, Alpha should obtain a minor gain on general codes. A slight additional improvement occurs in cases where Alpha does not require the explicit compare.

Several RS1 instructions do not exist in Alpha, one class of which is storage reference instructions that provide an implicit update of the address pointer. Rather than counting them as a path length advantage, we considered the pipeline effects of these implicit updates in the pipeline CPI effects by giving RS1 credit for a "virtual" functional unit. However, the update forms also provide some I-cache footprint reduction.

A second class of instructions is load multiple and store

***The distinguishing performance
characteristic of the Alpha
systems is clock rate; the
distinguishing characteristic of
the RS1 is support for
instruction-level parallelism.***

multiple—used primarily in subroutine prolog and epilog. Although implementations of these instructions could move multiple registers each cycle, RS1 moves one register (4 bytes) per cycle. Their strength comes from a single instruction replacing a moderate number of instructions. The resulting reduction in footprint increases the effectiveness of instruction caches, instruction buffers, and instruction issue bandwidth. Since a pair of these instructions replaces up to 64 load and store instructions, code density improvement can be significant.

The RS1 string operations provide similar I-cache benefits to the load multiple and store multiple instructions. Additionally, string operations allow block moves with no alignment requirements; RS1 handles all string operations in hardware.

The RS1 branch on count instruction replaces a loop-count decrement and a branch. Since its predominant use is in numerical applications, and it saves only one fixed-point unit instruction per loop iteration, its advantage will be minimized by the likely unrolling of floating-point loops for Alpha.

For floating-point code sequences, a major path length advantage of RS1 is the floating-point multiply-add (FMA) instruction. FMA instructions account for a significant percentage of the RS1 floating-point unit operations in SPECfp89, Linpack, Livermore loops, and many scientific/engineering applications. The compound multiply-add noticeably benefits some subset of floating-point codes.¹² Alpha requires two six-cycle floating-point unit pipeline passes to provide the function of the RS1 FMA ($A = B * C + D$) operation. Since the requirement for two passes stems from the need to use two instructions (multiply and add) to replace the compound instruction, this is a path length issue. Replacing the single FMA with a pair of dependent floating-point unit instructions does not increase the instruction-level parallelism. Therefore, if the FMA is on the critical path, doubling the path length of this compound operation doubles the corresponding cycle count.

The compound FMA instruction, along with the update forms of storage reference instructions, gives RS1 a clear path length advantage on floating-point applications.

Alpha architects rely heavily on compiler support to select instruction sequences that avoid performance pitfalls exposed by hardware simplifications. For example, Alpha's storage references can be grouped into two categories. Four-byte (8-byte) references that do not cross 4-byte (8-byte) boundaries can use normal loads and stores without penalty. For all unaligned references, the Alpha programmer has two choices. Using normal loads and stores is very slow, possibly generating warning messages to the user.⁶ The suggested alternative is special load-unaligned instructions that guarantee alignment by forcing the low-order address bits to zero. References to two adjacent elements are required to obtain unaligned 4-byte and 8-byte items. Additional instructions are required to extract or merge byte or 2-byte quantities. The compounding effect of alignment restrictions and lack of byte and 2-byte references is illustrated by the various instruction sequences shown in Alpha literature.⁶ The compiler must attempt to select the most efficient sequence. The best choice depends on the particular alignment, if known at compile time, and the storage reference access size.

In contrast to requiring instruction sequences for alignment and byte manipulation, RS1 loads (stores) often can be used without considering alignment. Frequently encountered types of unaligned references (unaligned 4-byte integer operands or 8-byte floating-point operands aligned on odd 4-byte boundaries) incur only one penalty cycle on RS1. Therefore, the path length of compiled code for these cases is a single instruction on RS1 whereas Alpha requires a sequence.

Evaluating performance

Vendors have reported SPEC Release 1.2 performance on many systems.¹³ Figure 2 shows some of these results, including labels that indicate the underlying architecture. (Since June 1992, SPEC has published higher results for many vendors. However, we don't intend Figure 2 to be an exhaustive set of competitive data. Data points are selected to illustrate the lack of correlation between clock rate and performance.)

Although performance tends to scale with clock rate for a given architecture, no clear relationship between clock rate and performance exists across architectures. The range of the SPECmark89 results at a given frequency, as well as the various on-chip frequencies required to deliver a given SPECmark89 result, clearly indicate that many factors other than frequency contribute significantly to performance.

Although we've discussed the magnitude of the performance impact for a few of the items in the two compared machines, we did not attempt to project overall performance. However, the overall impact of the performance effects we've

described can be demonstrated by the measurement data.

Table 3 lists the results for the two systems on various processor benchmarks.¹⁴⁻¹⁶ On SPECint92, the 160-MHz Alpha leads the 62.5-MHz RS1 by 50 percent; on the remaining benchmarks, the systems differ by less than 10 percent. As an average across these benchmarks, the RS1 performs at about 90 percent of an Alpha yet requires only 40 percent of the clock rate. Instruction-level parallelism is a practical alternative to aggressive clock rate for achieving performance.

While this Alpha system runs at more than two and one-half times the clock rate of the RS1, performance is comparable. The distinguishing performance characteristic of the Alpha systems is clock rate; the distinguishing characteristic of the RS1 is support for instruction-level parallelism. This difference in philosophies will become more pronounced as DEC continues to increase clock speed and IBM enhances superscalar capabilities, supporting multiple functional units, both fixed-point and floating-point. IBM PowerPC chips will merge the aggressive clock rate and superscalar approaches.

WE HAVE SELECTED ONE PROCESSOR to represent design opportunity available at 62.5 MHz and another to represent a 160-MHz design point. Based on the information currently available, we compared the performance aspects of the functional units, the interaction of the functional units, and the memory subsystem effects of these two processors.

We attempted to identify those details that were likely to be influenced by the designer's clock rate goal. Reviewing the trade-offs for the two processors showed that higher clocks are associated with longer pipelines; less instruction and operand queuing; less autonomy of the fixed-point and floating-point units; and smaller, simpler caches. The aggressive clock rate goal is also evident in the instruction set definition. The Alpha architecture document describes areas in which it relies on instruction sequences to replace single instructions that might be difficult to implement at higher clock rates, such as unaligned or byte storage references.

The major advantage for Alpha is its higher clock rate. A moderate CPI advantage of Alpha results from separate ALU and load and store units. However, this benefit is offset by dispatch restrictions, the virtual dual

fixed-point unit support provided by RS1 update forms, and Alpha's longer load use delays. A moderate path length advantage for Alpha stems from branches that incorporate simple tests, which, in some cases, remove the need for an explicit compare instruction.

The major CPI advantages of RS1 come from its FMA support; shorter pipelines; and larger, set-associative, store-in caches. Moderate CPI advantages of the RS1 come from early branch resolution, more flexible dispatch and issue, fixed-/floating-point unit instruction queues, and the pending store queue. RS1 gained moderate path length advantage with load multiple and store multiple instructions, which favorably affect the I-cache footprint.

In addition to clock rate, many factors contribute to measurable end-user performance and may be negatively affected by the clock rate and cost goals. The historical lack of correlation between clock rate and performance is best illustrated by a plot of SPECmark89 versus processor clock rate.

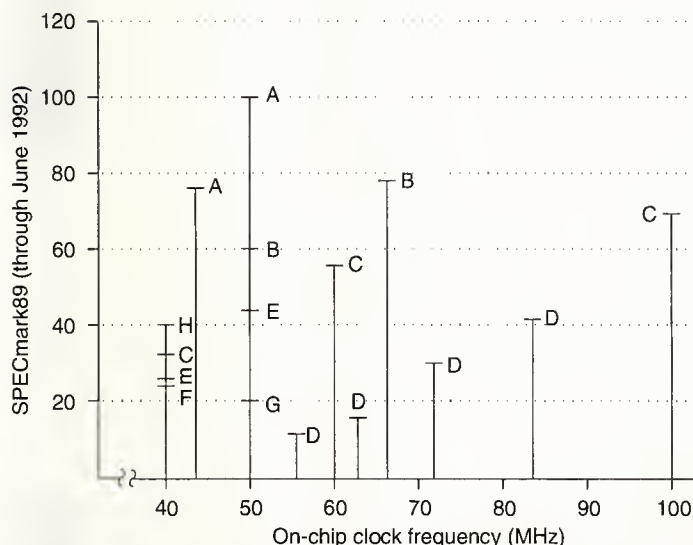


Figure 2. Cycle time is not enough to project performance. (Letters A-H indicate different instruction set architectures.)

Table 3. Benchmark measurements.

Benchmark	RS1 (62.5 MHz)	Alpha (160 MHz)	Ratio
SPECmark89	126.4	137.3	1.088
SPECint92	61.9	94.6	1.528
SPECfp92	134.6	137.6	1.022
Linpac(100x100)	38	36	0.947
Linpac TPP	104	114	1.096

We plan to continue evaluating the performance of industry-leading systems. Assuming sufficient reader interest, future articles will address follow-on systems. ■

Acknowledgments

We thank the processor and system designers at DEC and IBM for providing interesting design points to analyze. We also thank Sohail Saiyed for generating the traces, running the cache model, and modeling the Alpha write buffers; Juho Tang for modification of the cache model; and Rich Oehler, Kaivalya Dixit, Chuck Moore, and John Reysa for reviewing the article and providing many useful comments.

This article was submitted August 20, 1992, and revised for publication on August 20, 1993.

References

1. "IBM RISC System/6000 Technology," Order no. SA23-2619, IBM Corp. branch offices, 1990.
2. *IBM J. Research and Development*, Vol. 34, No. 1, Jan. 1990, entire issue.
3. *AIX Version 3.2 for RISC System/6000 Assembler Language Reference Manual*, Order no. SA23-2197, IBM Corp. branch offices, 1992.
4. "DECChip 21064-AA RISC Microprocessor Preliminary Data Sheet," Digital Equipment Corp., Maynard, Mass., Apr. 29, 1992.
5. Richard Sites and Richard Witek, "Alpha Architecture and First Implementation," *Proc. Compcon*, IEEE Computer Society Press, Los Alamitos, Calif., Feb. 27, 1992, p. 214.
6. *Alpha Architecture Handbook*, Digital Equipment Corp., 1992.
7. Daniel Dobberpuhl et al., "A 200-MHz, 64b Dual-Issue CMOS Microprocessor," *Proc. Int'l Solid-State Circuits Conf.*, IEEE CS Press, Feb. 20, 1992, pp. 106-107.
8. "DEC Enters Microprocessor Business with Alpha," *Microprocessor Report*, Mar. 4, 1992, pp. 1 and 6-14.
9. David A. Patterson and John L. Hennessy, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann Publishers Inc., San Mateo, Calif., 1990, p. 422.
10. "SPEC Benchmark Suite Release 1.0," *SPEC Newsletter*, Spring 1990, Vol. 2, No. 2, pp. 3-4.
11. *SPEC Newsletter*, Vol. 5, No. 2, June 1993, pp. 25, 28, 51, 54.
12. "IBM RS/6000's Complex Implementation Extracts Peak Performance," *Microprocessor Report*, Aug. 21, 1991, p. 14.
13. *SPEC Newsletter*, issues through June 1992.
14. Jack J. Dongarra, "Performance of Various Computers Using Standard Linear Equations Software," CS-89-85, Apr. 15, 1993, p. 6. (Postscript copies available through netlib@ornl.gov; in message, type "Send performance from benchmark.")
15. *SPEC Newsletter*, Vol. 4, No. 4, Dec. 1992, p. 62.
16. *SPEC Newsletter*, Vol. 5, No. 2, June 1993, pp. 26, 34, 52, 60, and 146.



Steven W. White is a senior engineer in IBM's AWS Processor Architecture and Performance Group. His primary interest is scientific and engineering performance. He joined IBM to work on mainframe scientific and engineering processor development, architecture, and system design.

White received his BSEE, MSEE, and PhD degrees from Texas A&M University, where he also taught for three years. He has published papers in a variety of areas including VLSI design, parallel processing, numerical algorithms, code optimization, system design, performance analysis, human genome research, and compilers. He is a member of the IEEE and the IEEE Computer Society and is a registered professional engineer.



Phil D. Hester is vice president, systems and technology for Advanced Workstations & Systems (AWS) in Austin and a member of the AWS Business Council. He holds responsibility for advanced systems and technology development of IBM RISC System/6000 products. Previously, he was responsible for development of the RS/6000 family and has been involved in the architecture, performance analysis, design, and implementation of the RISC-based products in IBM.

Hester received a BS degree in electrical engineering and an MS degree in engineering from the University of Texas at Austin. He is a member of Tau Beta Pi and Eta Kappa Nu, holds numerous patents, and has authored technical publications on RISC processors.



Jack W. Kemp, manager of Hardware Architecture, Advanced Workstations and Systems, is currently responsible for both processor and I/O architecture for the RISC System/6000 family of products. He managed the SCSI subsystem development effort for the original RISC System/6000 product and has held responsibility for processor architecture and performance.

Kemp holds a BS in solid-state physics from Marist College in Poughkeepsie, New York.



G. Jeanette McWilliams is an independent consultant who specializes in processor and memory subsystem performance. Previously, she worked at IBM as a senior programmer and hardware performance analyst on the RS/6000 team. She projected RS1 performance on industry-standard benchmarks prior to chip fabrication. She was IBM's first technical representative to SPEC and served on the Steering Committee for two years. Other performance assignments at IBM include transaction processing system support on mainframes and PC software performance assurance.

McWilliams received a BA degree in mathematics and an MA degree in computer science from the University of Texas at Austin and has completed IBM's Systems Research Institute. She is a member of the Association of Computing Machinery and the Independent Computer Consultants Association.

Direct questions concerning this article to Steven W. White, ZIP 9440, IBM Advanced Workstations and Systems, 11400 Burnet Road, Austin, TX 78758; or steve@hwperform.austin.ibm.com. McWilliams may be reached at wk00196@worldlink.com.

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 162

Medium 163

High 164

IEEE COMPUTER SOCIETY PRESS

2nd IEEE WORKSHOP ON PROGRAM COMPREHENSION

July 8-9, 1993 — Capri, Italy

WPC '93 deals with the dynamic new field of program comprehension and includes 20 reports on its theories, models, methods, and techniques. The text presents a comprehensive overview of current research in software engineering involving tasks such as maintenance, reuse, and reengineering.

Sections: Models and Proposals for Program Comprehension, Experiments in Program Comprehension, Experience Report Session, Program Representations, Integrating Documents and Codes, Tools for Program Comprehension.

208 pages. June 1993. Hardcover. ISBN 0-8186-4042-1.
Catalog # 4042-02 — \$64.00 Members \$32.00

1993 WORKING CONFERENCE ON REVERSE ENGINEERING

May 21-23, 1993 — Baltimore, MD

A research-oriented forum on the theory and technology of recovering information from existing software and systems. The text consists of 23 papers exploring innovative methods of extracting and using many kinds of information that can be recovered from software.

Sections: Reverse Engineering: Issues and Approaches, Identifying Components, Program Understanding, Challenges, Database, Tools, User Interfaces, Dynamic Analysis and Testing, Transformation.

248 pages. May 1993. Softcover. ISBN 0-8186-3780-3.
Catalog # 3780-02 — \$60.00 Members \$30.00

2nd INTERNATIONAL WORKSHOP ON SOFTWARE REUSABILITY

"Advances in Software Reuse"

March 24-26, 1993 — Lucca, Italy

IWSR-2 focuses on the theoretical and practical solutions to the software reuse problem. It explores the methods and tools for reuse solutions, and contains 20 papers addressing the critical areas of reuse design, measurement and metrics, and human factors, spanning the range from theory to practice.

Sections: Design, Libraries and Environments, Measurement and Metrics, Experience Reports and Human Factors.

216 pages. Softcover. ISBN 0-8186-3130-9.
Catalog # 3130-02 — \$60.00 Members \$30.00

To order call toll-free:

1-800-CS-BOOKS

FAX: (714) 821-4641

E-Mail: cs.books@computer.org



Software Report



David K. Kahaner

US Office of Naval

Research, Far East

kahaner@cs.titech.ac.jp

Cooperation: Japan's new watchword?

According to a report recently released by Japan's Management and Coordination Agency, that country spent over 13 trillion yen (US\$100 billion) on research and development in 1990, 10.7 percent above the previous year. Japan's research and development expenditures accounted for 3 percent of its GNP, a new record. The private sector accounted for about 11.7 trillion yen, 82 percent of overall R&D expenditures.

Reflecting the growing awareness of the importance of environmental protection worldwide, expenditures for environmental protection surged 20.3 percent over the previous year to 237.8 billion yen. Japan's exports of technology (receipts from patents and royalties) increased 3 percent over the previous year to 339.4 billion yen. Imports of technologies increased to 372 billion yen, 12.7 percent over the previous year. Japan's imports of technology from the US outpaced exports by 2.5 times.

As Japan attempts to keep pace with developments in the ever-changing world of high technology and computer science, its government and industry leaders are moving on several fronts simultaneously. As I describe in the snapshots that follow, efforts are under way to improve coordination and cooperation between the public and private sector there, as well as with foreign entities. Indeed, cooperation as much as competition appears to be the watchword in many of these efforts—cooperation between government agencies and private industry, with foreign businesses on either side of the Pacific, and between governments.

Stronger regional technical centers

The Ministry of International Trade and Industry's Agency of Industrial Science and Tech-

nology (AIST) plans to develop new regional technology policies and strengthen the research functions of the seven regional Government Industrial Research Institutes (GIRI) under AIST's umbrella in fiscal year 1993. The policies are designed to correct the over-centralization of industry. At the same time, upgrading unique R&D bases should prove indispensable to invigorating the regions. AIST aims to actively develop the various regional research resources centering on the GIRIs, marshal the public testing and research institutes (*kobsetsushi*), third-sector research centers, and private companies to play a guiding role in the regional development, and link them with local universities.

The importance of exchanges between GIRIs and *kobsetsushi* centers has been stressed before, but the various players have not yet conducted exchanges in a sustained fashion. AIST will support research by arranging tie-ups between *kobsetsushi* centers, private companies, universities, and other organizations. Changing *kobsetsushi* centers into industrial technology centers and renovating facilities and systems is moving ahead. By linking the centers to local universities, the policy will open the way for researchers at GIRIs to earn doctorates based on research achievements there.

ASIC chip makers mount challenge

Japan's semiconductor makers, faced with a mounting offensive on their home turf by US companies, are hustling to build up a presence in application-specific integrated circuits, in some cases by cooperating with US companies. ASIC chips have come into their own as semiconductor users move away from multipurpose chips, but they also have assumed heightened importance in light of the market penetration by US

chip makers.

Demand is growing for programmable logic devices and field-programmable gate arrays from system developers looking for devices to incorporate high-speed systems that outstrip the capacity of conventional gate arrays. The programmable logic market amounted to US\$900 million in 1991, only around 25 percent of the size of the gate array market. According to industry estimates it should expand to around US\$2.2 billion by 1995.

Disc makers swap patents

Hoping to avoid another bitter confrontation over two competing technologies, Matsushita Electric Industrial Co., Sony Corp., and Philips Electronic BV reportedly have concluded basic agreements to make their patents on digital compact cassettes and minidisks mutually accessible. The two Japanese manufacturers have led rivals in the commercialization of the technologies—small digital recording devices that generate high-quality sound. Matsushita adopted the cassette format, while Sony picked the disc format. Philips has committed itself to both types of equipment.

Matsushita, Sony, and Philips will control their patents uniformly, supplying technology to domestic and overseas enterprises interested in producing both hardware and software for digital compact cassettes and minidisks. Mutual access to the patents on compact cassettes and minidisks will allow the two Japanese camps to hone technology applied to their products, company officials say. Meanwhile, the agreement will pave the way for Matsushita to penetrate Sony's turf and vice versa. Matsushita has begun to market its first digital compact cassette recording, while Sony released its minidisk recorder at the end of 1992.

Construction industry discovers satellites

Japan's construction industry has joined the growing contingent of busi-

nesses worldwide that have turned to the heavens for guidance—not to the stars, mind you, but to the satellites that make up the Global Positioning System (GPS). The GPS is a group of satellites launched by the US Department of Defense that was originally intended for boat and aircraft navigation. At present, 18 GPS satellites orbit 20,000 km above earth. Three more are scheduled for launch by the end of 1993, bringing the total to 21, or enough to provide 24-hour access anywhere in the world. While car navigation systems form the highest profile commercial GPS application, the construction industry is learning to make use of the GPS satellites for land survey work as well.

Surveying is simple with GPS—all you need is a pair of receivers, antennas, and three satellite signals. Measurements of the distance between the two receivers and the elevation at the site of second receiver can reach an accuracy of 0.0001 percent. The GPS system eliminates the traditional equipment surveyors use to measure distance and angles.

The first commercial GPS system came out in 1986, and already this method has become the standard tool for many survey applications in the US. Europe was also quick to make use of the system in 1987, but Japan has lagged a bit. Of the roughly 40,000 GPS survey receivers that have sold worldwide, only 250 reportedly are in operation in Japan. However, several surveying firms there are now using the system, and more are likely to follow.

"The domestic market for GPS survey receivers finally began to take off last year," says Hideyuki Torimoto of Trimble Navigation Systems, an American firm that leads the world in the manufacture of GPS receivers. As recently as two years ago, GPS receiver sets cost as much as 15 million yen (\$120,000). Trimble now markets a model for general precision survey work that costs only 5.9 million yen.

Japan, South Korea, Europe cooperate on fast ISDN

These traditional competitors agreed recently to jointly develop a large-scale communications network that would transmit data among countries at high speed, according to government officials in both countries. The accord came during a regular conference between the Japanese Posts and Telecommunications Ministry (MPT) and its South Korean counterpart in Seoul. The integrated service digital network (ISDN) would permit transmission of different communications services, including digital telephone calls, facsimile transmission, and data communications, on a single network.

Many countries now are studying the possible commercial operation of the new system. However, because ISDN system transmission modes vary from nation to nation, some countries have found it impossible to exchange data and information. To standardize ISDN, Japan and the European community are scheduled to experimentally connect their communications lines next year. This agreement will help to standardize communications modes in the Asia-Pacific basin. To promote their joint project, Japan and South Korea soon are likely to select the participating communications enterprises and communications equipment manufacturers.

Computer downsizing trends

Japanese computer makers, which have long challenged IBM's supremacy in mainframes, are now following that company's lead in the opposite direction. Like IBM, they are reducing their dependence on large computers in favor of system planning and maintenance services. But the pace of the shift away from big, costly computers to a network of smaller machines has been slow compared with the US. Here, the downsizing trend has been so rapid that IBM has predicted zero growth in hardware revenue and other firms have abandoned the field altogether.

The story is different in Japan. There, though down from 70 percent in 1989, mainframes still hold 60 percent of the market, and smaller machines are suffering from the sluggish economy. Yet many observers say that change is inevitable. The limited availability of Japanese language software for networks of small computers has slowed the pace, they say. Also, the greater variety of computers sold in Japan, making networking more difficult, has impeded progress. Industry estimates put the percentage of personal computers and workstations forming networks at less than 10 percent in Japan, whereas the number is reportedly between 30 and 40 percent in the US.

According to Shozo Shigeoka, editor-in-chief of *Nikkei*, a leading computer magazine, "department heads of many Japanese companies make decisions after discussing subjects with all bosses up the ladder. Distributed computing does not fit well with this centralized decision-making style." Everybody wants to look at all the information, he says, while lower ranking western managers have more decision-making power. That, combined with the substantial mainframe software assets they have accumulated, has made Japanese companies less eager to try new systems, according to Shigeoka.

IBM seeks Canon's help

IBM has added its personal computer operations to the growing list of areas in which it has sought the help of a prominent Japanese manufacturer. IBM and Canon have agreed to cooperate in the development of desktop and portable computers. This news coincided with an announcement by IBM that its PC development, manufacturing, distribution, and marketing operations would be consolidated in a new, autonomous unit known as the IBM Personal Computer Company.

A Canon spokesman says that one of the first tasks of the new alliance will be to develop a portable PC with a built-in printer. The partnership will

help IBM tap Canon's expertise in computer peripheral equipment, especially printers. IBM already has joined forces with Toshiba and Hitachi to develop advanced semiconductor chips and high-end printers. Canon possesses technology in color flat-panel displays and optical magnetic disks that also could be of interest to IBM.

Scientific research is growing outside Tokyo

Scientific and technological activities in regions outside Tokyo are increasing, according to a recent report from the Science and Technology Agency (STA). That organization found that research institutes, led by those in the private sector, are moving increasingly into regions outside of Tokyo and its three surrounding prefectures, Kanagawa, Saitama, and Chiba. The report also called for harmonizing national and local science and technology policies, increasing the communication between the national and local governments, and improving the quality of regional science and technology.

According to the STA white paper, corporations in the private sector are accelerating regional scientific and technological activities as they shift their research centers to outside the metropolitan region. The report noted that research institutes of private firms and public organizations have been moving into regions outside the Kanto area at a considerable pace for the past several years. They tend to concentrate in a limited number of areas near major cities in each prefecture.

Noting that in recent years the private sector has established very few sizable research facilities in Tokyo, the report says that the Kanto area saw its share of newly opened research institutes sharply reduced to 34 percent for the 1989-1991 period, down from 52 percent in the previous three-year period.

Touching on growing international cooperation in scientific and technological research, the report says, "it is possible for local authorities to join

**A Canon
spokesman says
that one of the
first tasks will be
to develop a
portable PC with
a built-in printer.**

hands with the national government" in carving out policies in some fields. It cites the difficulty of securing personnel and providing an "adequate living environment" for researchers in regions outside Tokyo. The report notes the need to improve the quality of regional science and technology, as well as to work jointly on policies and facilitate communication between the national and local individuality, and comprehensive policy-making.

"The national government cannot dishearten regional independence," the report concludes. "Our work suggests that local government develop advanced science and technology, adding national policies to regional potentials."

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 195 Medium 196 High 197

New Products

Joe Hootman

University of North Dakota

Communications/displays

Communicator combines functions

Developed jointly by AT&T and Eo, Inc., the 440 Personal Communicator combines fax, electronic mail, personal productivity applications, and cellular phone capabilities. Available at AT&T Phone Center Stores, this handheld, pen-based device comes in a variety of configurations, including one equipped with an 8-Mbyte RAM, 20-Mbyte hard drive, and internal modem. *AT&T; from \$1,999.*

Reader Service No. 10

Power-stingy fax modems

PCMCIA-based fax modems offer battery-saving ultra-low-power mode, flash memory for on-line operating program upgrade, and built-in DAA line interface circuits. Operating at 50 mA and dropping below 2 mA in an on-command sleep mode, the SmartExchange 1414 provides data and facsimile capabilities, error correction, and data compression in a credit-card-size format. Offering full-duplex data communication at 14.4 Kbps, the 1414 permits four-to-one data compression, giving an effective throughput of up to 57.6 Kbps.

Entry-level 9624 and 9624E models operate at 2.4 Kbps for full-duplex communications; both come with standby and sleep modes. The "E" version accommodates built-in high-level error correction and data compression. *Smart Modular Technologies; \$549.99 (1414), \$399.99 (9624E), \$299.99 (9624).*

Reader Service No. 11

Ethernet LAN, Token Ring adapter cards

Sixteen-bit I/O map ISA bus Ethernet cards boost throughput by using I/O and memory mapping modes to issue early interruptions so packets transfer quickly to host computer memory. Built for PC/XT/AT bus ISA-compatible sys-

tems, these LAN adapters work well with Novell's NE2000 Ethernet card and comply with Ethernet 802.3. The jumperless CN888E lets users fine-tune performance by configuring the I/O address, IRQ channel, boot ROM, fast read, CHRDY control, and I/O16 control. Both CN200E and CN600E come with a 16-Kbyte data packet buffer and onboard 8-Kbyte PROM socket for adding a remote-boot ROM.

Also available for 16-bit performance is the CN2000T Token Ring adapter. Supporting 4- or 16-Mbps network speed and offering both 8- and 16-bit data transfer, this AT- and ISA-system compatible card accommodates either STP or UTP wiring. *CNet Technology; \$299 (CN888E), \$99 (CN600E, CN200E), \$439 (CN2000T).*

Reader Service No. 12



CNet CN2000T Token Ring adapter

LIU cuts digital transmission costs

Functionally compatible with most industry-standard single-channel transceivers, the VP14Q574 line interface unit gives users power dissipation advantages compared to discrete T1/E1 LIUs. Needing a single external crystal, it supports both DSX-1 (ANSI) or E1/CEPT (CCITT) formats for channel banks, multiplexers, office repeater relays, digital cross-connection systems, and digital switches. An 8-bit microprocessor interface, common to all four transceivers, provides extensive monitoring and control capabili-

ties. Packaged in a 128-pin plastic QFP, its features include group selection of jitter attenuation direction and internal multiplexers for nonintrusive monitoring of eight transmit/receive data paths. *VLSI Technology; \$42 (10,000s).*

Reader Service No. 13

Handheld protocol analyzer

A PC-based protocol analyzer for notebook computers, the ParaScope 64M operates as a stand-alone device that does not require a card slot. Communicating via standard 4- or 8-bit parallel ports, this compact device running Feline software handles a wide range of protocols, achieving a throughput rate of better than 64 Kbps on a 25-MHz 386 machine. Available data codes include DDCMP, CRC-CCITT, CRC-16, CR-12, CRC-6, LRC, and parity error checking. Packaged in a 9.22x4.76x1.97-inch molded plastic case, PS6145 works off both AC and battery power, and comes with a battery-saving low-power mode. *Fredrick Engineering; \$3,295.*

Reader Service No. 14

Multiline voice, fax processing

Voice Ranger, a multiline voice processing board, features two ports per card and permits expansion to 16 ports per system. Using the device's 16-Kbits of memory per card, users can create multitasking voice applications under DOS, directly record voice files, and simultaneously record incoming speech on multiple lines. With the complementary Commando Developer's Tool Kit, a multitasking TSR operating in the DOS environment, developers can create fax mail, audiotex, voice bulletin boards, fax-on-demand, and order entry with credit card authorization applications.

V/S Plus, a multiline voice and fax processing package, includes a multiline voice card, plus voice- and fax-processing software. Offering fax-on-demand with most CAS-compatible fax cards, V/S Plus accommodates multi-level call processing applications and

1,000 voice mail boxes. Also, Faxmouth, a fax-on-demand add-on to the Bigmouth single-line voice processing system, lets callers leave voice mail messages and request faxed documents in a single phone call, retrieve stored documents, and fax them to selected fax numbers. *Talking Technology; \$599 (Voice Ranger), \$169 (Commando), \$699 (V/S Plus), \$199 (Faxmouth).*

Reader Service No. 15

Tool kit for IBM LAN systems

The NDIS Driver Developer's Tool Kit gives programmers samples, specifications, test tools, and documentation needed to design and implement NDIS 2.0.1 media access controller (MAC) device drivers. Features include an NIF validation program, NDIS trace tool, and IBM Token Ring sample driver. *DWB Associates; \$575.*

Reader Service No. 16

RS-422, RS-485 support

For greater flexibility in implementing advanced, client-server systems without relying on Ethernet-based expansion, RS-422 and RS-485 support has been added for these standard SBus expansion boards. With RS-422, Sparc systems connect to dozens of terminals, modem banks, and other peripherals at up to 4,000 feet. The multidrop capability of RS-485 works well for applications requiring numerous peripheral connections along a single serial line. Compatible with CCITT V.10/11 and X.26/27, the four-port module meets EIA standards. *Aurora Technologies; \$499.*

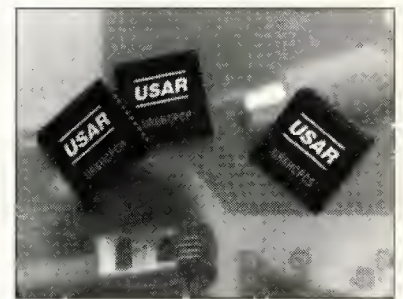
Reader Service No. 17

Protocol converters

Providing up to 9,600-bps serial transmission speed, the single-chip UR6HCPCX HCMOS protocol converter links PC-compatible user input devices to hosts having a serial or parallel port. By providing all electrical and functional features of an 8042 system keyboard controller, plus all tiers

of keyboard BIOS control, it matches performance formerly obtained through AT/PS2 motherboard connections. Housed in 40-pin DIP, 44-pin PLCC, or QFP packages, the converter presents data to the host in either PC-standard scan code format or translated extended ASCII codes. *USAR Systems; \$27.50 (100s).*

Reader Service No. 18



USAR UR6HCPCX protocol converters

QUICC chip controller card

Dubbed the VCOM-54, this VMEbus communication controller uses Motorola's 32-bit 68360 Quad Integrated Communications Controller to provide four or eight ports of T1 or E1 speed connectivity through its P2 or front panel connectors. The controller features 1 or 4 Mbytes of parity-protected DRAM, VME 32/64-bit compatibility, VME DMA, and 68020 software compatibility. *SBE; from \$2,385 (100s).*

Reader Service No. 19

Token Ring hub

An intelligent 16-port Token Ring hub, the TokenEase Smart MSAU lets network managers control ring-in/ring-out loopback, monitor port activity status, set password security, print reports, and view the port connection status via an on-screen worksheet. Controlled from a remote PC via out-of-band software management and through an RS-232 port, TokenEase features fault-tolerant ring-in and ring-out in case the trunk connection is broken, auto-loop back for preserving ring integrity, and LED diagnostic indicators

for troubleshooting. Included software can control 256 daisy-chained units and features a scenario function for downloading multiple network configurations. *MUX Lab*; \$1,750.

Reader Service No. 20

Compact Token Ring MAU

Designed to fit a standard 19-inch rack, but only 1-inch deep, the 8228-equivalent Compact MAU fits smaller wiring closets or in-office locations. With eight lobe ports plus ring-in/ring-out, the unit's global reset button simultaneously initializes all lobe ports. Adjacent ring-in/ring out ports enable cleaner rack-style cabling. *Belkin Components*; \$499 (discounts available).

Reader Service No. 21

Switchmode control IC

Operating directly from telephone line voltages, the Si9114 enables operation at switching frequencies beyond 500 kHz, allowing for use of smaller magnetics and filters, and eliminating the need for electrolytic capacitors. Featuring soft-start, internal start-up, and latched shutdown circuitry, the 14-pin device can implement either a flyback or a forward converter at switching frequencies of 1 MHz. Available in plastic DIP and SOIC packages, its synch output pin allows additional power converters synchronized to each other or to an external clock, both in phase and frequency, simplifying EMI filtering.

Planar transformer techniques reduce external parts count and circuit board area, making for an 8-mm board thickness. *Silconix*; \$1.73 (OEM quantities).

Reader Service No. 22

Color flat panel display

Designed for use in industrial controls, test and measurement instruments, and medical equipment, this EL display offers wide-angle viewing and a wide operating temperature range. Displaying eight colors on a high-con-

trast black background, the EL640.350-DA1 uses a nonreflective structure that needs no contrast-enhancing filters. For easy mounting, the EL glass panel and control electronics are assembled into a 226x153x20-mm package. Typical power consumption is 16W. *Planar Systems, Inc.*; \$2,495 (each), \$1,210 (1,000s).

Reader Service No. 23



Planar Systems' EL640.350-DA1

Touch screen controller

Packaged in a plastic box for mounting on the back or bottom of a CRT, this CMOS touch screen controller simplifies monitor refitting by avoiding internal mounting complications. A pocket-sized 3.75x2.5x0.9 inches, the SMT-1 operates off 70 mA of current and requires a single +5V power supply (or from any voltage from +8 to +16 VDC). The controller achieves a MIL-HBK-217-F1 MTBF rate of 572,600 hours. *MicroTouch Systems, Inc.*; \$318 (volume discounts available).

Reader Service No. 24

Low-radiation monitors

Offering 0.28-mm dot pitch and 1,280x1,024-pixel resolution, the ASTVision low-radiation, multi-sync color monitors come in 14-, 15-, and 17-inch formats. Push-button, digital controls allow for horizontal and vertical sizing and positioning. Featuring 10 preset or eight custom modes for changing display configurations in a Windows-based environment, the monitors come with recyclable bromide-free plastic cases. *AST Research*; \$495 (14-inch), \$595 (15-inch), \$995 (17-inch).

Reader Service No. 25

DSP components

Parallel processing boards

Two DSP boards provide support for eight 50-MHz TMS320C40 DSPs and up to 400 Mflops in a single PC or VME slot. Featuring a scalable topology that enables addition of TIM-40 standard single and twin processor modules and boards, C40 Octal Processor Systems can be used to build large networks for use in advanced imaging, sonar, radar, simulation, and modeling system applications. Offering direct host access to each processor through the JTAG interface, the systems allow code debugging in C, assembler, or both simultaneously. Accompanying software tools include an assembler/linker, ANSI C compiler, C source-level debugger, 3L parallel C compiler, Virtuoso, and FloTar and FasTar libraries. *Spectrum Signal Processing*; \$19,145 (PC), \$18,240 (VME), volume discounts available.

Reader Service No. 26

Multi-DSP board for VMEbus

Well suited for high-speed results in image processing, radar and sonar signal analysis, telemetry and telecommunications, simulation, and aerospace applications, the DSP-4 board moves data at over 1.2 Gbytes/s and processes at more than 300 Mflops. Built around three TMS320C40 processors, each board boasts 1 Mbyte of no-wait private RAM space, expandable to 16 Mbytes, and a 4-Mbyte block of shared memory available to all on-board processors. Included are the SPOX library of DSP routines, with FFT, other array and transform operations, FIR and IIR filters, and transcendental functions. Requiring 60W of 5V power, the DSP-4 fits a single 6U slot of a standard VMEbus card cage. *Pacific Cyber/Metrix*; \$15,349.

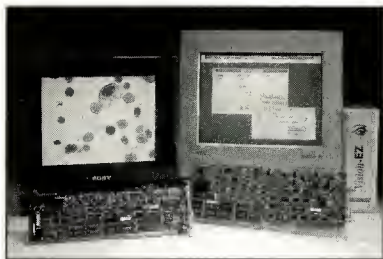
Reader Service No. 27

Data acquisition boards

Offering on-board simultaneous sample-and-hold and DSP intercon-

nection, FAST Series boards operate at a 1-MHz peak channel sample rate and 250 kHz/channel across four channels. Suited for applications requiring measurement of multiple sources of simultaneous events such as automated test fixtures and medical data acquisition, the SSH version features four channels of SSH resident in a single PC/AT slot, enabling implementation of DAS applications that require 12-bit resolution. The DSP version comes in either DSP-Link or DT-Connect; both feature a daughterboard containing a 4-Ksample FIFO buffer. The SSH version can also include the DSP option. Vibration analysis, speech processing, and spectral analysis are typical uses that require high-performance data acquisition coupled with DSP for real-time data reduction and analysis. *Analagic; from \$2,995.*

Reader Service No. 28



Analogic SSH/DSP DACs

High-speed DSP chip set

A 25-MHz, 24-bit fixed-point DSP, the LH9124LY processes 8- to 24-bit data in real time and performs digital filtering, image recognition, compression, spectrum analysis, correlation, convolution, and adaptive filtering in the time and frequency domains. Featuring six on-board multiplier/accumulators, eight adders, four complex bidirectional buses, and 24-bit external and 64-bit internal precision, the chip set handles very large arrays, 2D arrays, or data from up to 32 independent channels. Supported by the LH9320LU-25 address generator, the 26-function device performs a radix-16 butterfly in one pass and a 1K complex FFT in 129 μ s. Built using a

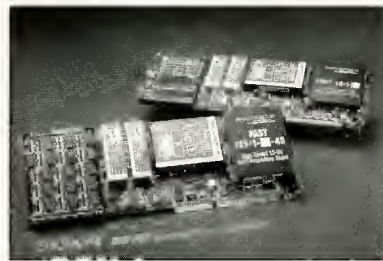
0.8- μ m, double-metal CMOS process and packaged in a 262-lead PGA, three cascaded LH9124LYs perform a complex 1K FFT in 41 μ s. Typical applications include ultrasound, spectrum analysis, and tomography. *Sharp Electronics; \$760 (100s, LH9124LY-25), \$125 (100s, LH9320LU-25).*

Reader Service No. 29

Image acquisition

Vision-EZ, a Windows-based image acquisition package, uses a frame grabber and software to display live video and capture, save, and print images. Using 640x480 square pixel spatial resolution and 256 gray levels, Vision-EZ captures images in real time from video cameras, VCRs, and still-video devices. Accompanying Global Lab Acquire software lets users save images in TIFF, PCX, or DT-IRIS formats. With four on-board image-enhancing input LUTs, the board handles arithmetic, contrast adjustment, and reverse video functions. *Data Translation; \$995.*

Reader Service No. 30



Data Translation's Vision-EZ

Analog input card

An 18-bit data acquisition system for the PC/104 bus, the 4A22 boasts 16 conversions per second, with an integration time that can be set to maximize 50- or 60-Hz rejection. Featuring software-selectable 0.5V or 5V full scale analog input ranges and 14 input channels with 500V isolation, the 4A22 stores calibration data in an on-card EEPROM, eliminating the need for zero- or full-scale adjustment potentiometers. For maximum accuracy, an on-card temperature readout channel

allows for software compensation of reference voltage drift, while galvanic isolation of input signals eliminates common ground loop problems. Driver software for background data collection is included. *Mesa Electronics; \$171 (100s).*

Reader Service No. 31

CAD tools

ASIC standard cell library

Using designs generated with the AnaCMOSLib V5.0 layout library, manufacturers can fabricate chips in 2- to 0.8- μ m, double-metal CMOS-bulk technologies. This analog/neural network ASIC layout library upgrade supports AMI, ES2, Hewlett-Packard, Orbit, US2, and VLSI vendors, and can run on IBM-PC and compatible, Sparc, HP-9000, and Macintosh systems. New elements include transmission gate multiplexer; transmission gate with active high, with active low enable, and with complementary enables; poly resistor; and N- and P-channel FETs. *Tanner Research; from \$495 (DOS systems).*

Reader Service No. 32

Thermal modeling for Pentium

Using CFD techniques to characterize thermal behavior, Flotherm software helps designers package Intel's Pentium, which dissipates up to 16 watts of heat, almost double its predecessors. Based on a 3D solution of equations governing airflow around components as well as conduction within, Flotherm models help define new system cooling requirements at an early development stage. *Flomerics; yearly licenses from \$11,000.*

Reader Service No. 33

Integrated design environment

ASIC Powertools incorporates various design tools in one package for VHDL design, timing analysis, and test applications. An integrated multi-methodology design flow comprised



October 1993 issue (card void after April 1994)

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone(_____) _____

Reader Interest

(Add comments on the back)

Readers,
Indicate your interest in
articles and departments
by **circling the appropriate number** (shown on
the last page of articles
and departments):

150 151 152 180 181 182
153 154 155 183 184 185
156 157 158 186 187 188

159 160 161 189 190 191
162 163 164 192 193 194
165 166 167 195 196 197

168 169 170 198 199 200
171 172 173 201 202 203
174 175 176 204 205 206

177 178 179 207 208 209

Product Information

(Circle the numbers to receive
product information)

1	21	41	61	81	101	121	141
2	22	42	62	82	102	122	142
3	23	43	63	83	103	123	143
4	24	44	64	84	104	124	144
5	25	45	65	85	105	125	145
6	26	46	66	86	106	126	146
7	27	47	67	87	107	127	147
8	28	48	68	88	108	128	148
9	29	49	69	89	109	129	149
10	30	50	70	90	110	130	
11	31	51	71	91	111	131	
12	32	52	72	92	112	132	
13	33	53	73	93	113	133	
14	34	54	74	94	114	134	
15	35	55	75	95	115	135	
16	36	56	76	96	116	136	
17	37	57	77	97	117	137	
18	38	58	78	98	118	138	
19	39	59	79	99	119	139	
20	40	60	80	100	120	140	



October 1993 issue (card void after April 1994)

Name _____

Title _____

Company _____

Address _____

City _____ State _____ Zip _____

Country _____ Phone(_____) _____

Reader Interest

(Add comments on the back)

Readers,
Indicate your interest in
articles and departments
by **circling the appropriate number** (shown on
the last page of articles
and departments):

150 151 152 180 181 182
153 154 155 183 184 185
156 157 158 186 187 188

159 160 161 189 190 191
162 163 164 192 193 194
165 166 167 195 196 197

168 169 170 198 199 200
171 172 173 201 202 203
174 175 176 204 205 206

177 178 179 207 208 209

Product Information

(Circle the numbers to receive
product information)

1	21	41	61	81	101	121	141
2	22	42	62	82	102	122	142
3	23	43	63	83	103	123	143
4	24	44	64	84	104	124	144
5	25	45	65	85	105	125	145
6	26	46	66	86	106	126	146
7	27	47	67	87	107	127	147
8	28	48	68	88	108	128	148
9	29	49	69	89	109	129	149
10	30	50	70	90	110	130	
11	31	51	71	91	111	131	
12	32	52	72	92	112	132	
13	33	53	73	93	113	133	
14	34	54	74	94	114	134	
15	35	55	75	95	115	135	
16	36	56	76	96	116	136	
17	37	57	77	97	117	137	
18	38	58	78	98	118	138	
19	39	59	79	99	119	139	
20	40	60	80	100	120	140	

SUBSCRIBE TO IEEE MICRO

All the facts about today's chips and systems

☐ YES, sign me up!

If you are a member of the Computer Society or any other IEEE society, pay the member rate of only \$23 for a year's subscription (six issues).

Society: _____

IEEE membership no: _____

Society members: Subscriptions are annualized. For orders submitted March through August, pay half the full-year rate (\$11.50) for three bimonthly issues.

Full Signature _____ Date _____

Name _____

Street _____

City _____

State/Country _____ ZIP/Postal Code _____

☐ YES, sign me up!

If you are a member of ACM, ACS, BCS, IEE (UK), IEEE but not a member of an IEEE society, IECEI, IPSJ, NSPE, SCS, or other professional society, pay the sister-society rate of only \$42 for a year's subscription (six issues).

Organization: _____ Membership no: _____

For information on airmail option, send fax request to (714) 821-4010.

☐ Payment enclosed *Residents of CA, DC, Canada, and Belgium add applicable sales tax.*

☐ Charge to ☐ Visa ☐ MasterCard ☐ American Express

Charge-card number _____

Expiration date _____

Month _____ Year _____

Prices valid through 12/31/93
1093 MICRO

Charge orders also taken by phone:
(714) 821-8380 8 a.m. to 5 p.m. Pacific time
Circulation Dept.
10662 Los Vaqueros Circ., PO Box 3014
Los Alamitos, CA 90720-1264

Editorial comments

I liked: _____

I disliked: _____

I would like to see: _____

Reviewers Needed. If interested, send professional data to Dante Del Corso, Dipartimento di Elettronica, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129, Torino, Italy.

For reader service inquiries, see other side.



PLACE
POSTAGE
HERE

PO Box is for reader service cards only.

IEEE Micro

PO BOX 16508
NORTH HOLLYWOOD CA 91615-6508
USA



Editorial comments

I liked: _____

I disliked: _____

I would like to see: _____

Reviewers Needed. If interested, send professional data to Dante Del Corso, Dipartimento di Elettronica, Politecnico di Torino, C.so Duca degli Abruzzi, 24, 10129, Torino, Italy.

For reader service inquiries, see other side.



PLACE
POSTAGE
HERE

PO Box is for reader service cards only.

IEEE Micro

PO BOX 16508
NORTH HOLLYWOOD CA 91615-6508
USA



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 38 LOS ALAMITOS, CA

POSTAGE WILL BE PAID BY ADDRESSEE

IEEE COMPUTER SOCIETY

PO BOX 3014
LOS ALAMITOS CA 90720-9804
USA



of graphics- and language-based design entry, synthesis, multilevel VHDL and structural simulations, timing analysis, test analysis, and ATPG, the suite of tools comes in two configurations: ASIC Architect and ASIC Expert. Tools include design composition, multilevel simulation, VHDL simulator, Motive integration for static timing analysis, and Sunrise test solution integration applications. *Viewlogic; from \$70,000.*

Reader Service No. 34

ASIC system design tools

A library of 1.0- and 0.8- μm gate array and embedded cell array families now supports the Synopsys VHDL System Simulator. Offering debugging and analysis tools to isolate and resolve coding errors, VSS V3.0 lets designers capture concepts, verify high-level specifications, and detect design inconsistencies before committing to gate-level implementation. Designed using the Synopsys Library Compiler, the design kit boasts 12K to 400K raw gates at the 0.8- μm gate level, and 6K to 70K raw gates for 1.0- μm designs. The 0.8- μm devices clock at up to 100 MHz, dissipate power down to 2.4 $\mu\text{W}/\text{MHz}/\text{gate}$, and run at 215 ps for a two-NAND gate. Available in plastic, ceramic, or power QFPs and 576-pin TAB packages, all gate arrays and embedded cell arrays feature a four-alternative speed/power option, selectable slew rate, and both 3V and 5V fully characterized cell libraries.

Also available is a design kit to support Mentor Graphics V8.2 for top-down design of these 0.8- and 1.0- μm arrays. Supported are Top Down Design-Solver, QuickFault II, FLMS, a Falcon Framework-based application for library development and management, and X terminal support. *Mitsubishi Electronics America.*

Reader Service No. 35

Yield-enhancement software

To help resolve questions about design-process mismatches that re-

duce yields, Data Mapper combines design, test, inspection, and process data in a statistically rich graphical environment. Communicating with scanning electron microscopes and focused ion beam equipment running Framework, this approach characterizes and images defects to help determine their source in the IC manufacturing process. For each set of defined wafers and variables, Data Mapper generates a display consisting of wafer maps, histograms, defect images, and the IC design layouts of each physical die layer. The Bit Mapper option automatically relates defect data and images to bit failures. Data Mapper imports data from SQL databases, tester-specific formats, ASCII files, and CAD data formats. *Knights Technology.*

Reader Service No. 36



Knights Technology's Data Mapper

VHDL design simulator

Developed jointly by Mentor Graphics and Model Technology is the QuickVHDL Simulation Family. QuickVHDL and its Pro Systems and Pro IC options offer direct compiled code simulators for the ASIC, IC, and system design environments. Integrated with Design Architect software for high-level graphics entry and automatic VHDL generation, the family also works closely with AutoLogic for synthesis and supports the Std_DevelopersKit. Supplied with design extractor, synthesis compiler, and cosimulation interface features, it com-

piles VHDL into generic RISC instructions, which then get mapped into the instruction set of the workstation that is executing the simulation. Designed for Sparc and HP-PA platforms, QuickVHDL also offers a Pro System configuration for accessing AMP-based ASIC libraries and board-level models with QuickSimII. The Pro IC option runs M models, gate-level Lsim primitives, switch-level models, SPICE models, and switched capacitor models via the Lsim simulator. *Mentor Graphics; \$19,950 (QuickVHDL), \$29,950 (QuickVHDL Pro System, Pro IC).*

Reader Service No. 37

Expanded CBiCMOS cell library

With 35 new cells, the RSC4000 mixed-signal cell library gives designers of video processing, instrumentation, and communications applications an expanded set of mixed analog and digital application-specific tools. Consisting of analog bipolar, CMOS logic, and data conversion cells, the library uses a 2- μm CMOS and complementary bipolar process with 4-GHz NPN and 1.5-GHz PNP transistors and thin-film resistors. Additions include CMOS logic gates, flip-flops, multiplexers, and I/O devices, as well as bipolar ECL logic cells. Typical design specifications include a 2-nV/root Hz noise performance, 200-MHz signal processing bandwidths, and 2,000-V/ μs slew rates. *Raytheon.*

Reader Service No. 38

Reader Interest Survey

Indicate your interest in this article by circling the appropriate number on the Reader Service Card.

Low 189 Medium 190 High 191

Product Summary

Joe Hootman

University of North Dakota

Manufacturer	Model	Comments	R.S.#
Boards			
Bit 3 Computer	Bus-to-bus adapters	RISC/Unix family's common VMEbus card and support software driver permit application software portability from Sparcstations, RS/6000s, HP 9000s, DECstation 5000 Series 700s, and the Silicon Graphics Indigo to the VMEbus. <i>\$2,850 each.</i>	80
Delta Computer Systems	MC 186/40 board	Improved four-axis motion control board offers fast, precise position control of hydraulic and electric servo systems that use the Multibus I or the Reliance AutoMate programmable controller. Coordinated position and speed changes can be made on the fly.	81
Microway	Quadputer-860 processor	Four i860 RISCs configured as a shared-memory MIMD device provide 200-Mflops numeric performance on EISA PCs. Five 25-MHz, 32-Mbyte Quadputers used in one EISA workstation provide an aggregate throughput of over 1 Gbyte (requires Unix). <i>\$11,955.</i>	82
Systems			
Tektronix	GPX/DAS logic analyzer	Logic analyzer with full disassembly capability and probe adapter supports Intel Pentiums on the general-purpose GPX analyzer for medium-size design tasks and the DAS 9200 systems analysis platform for multiprocessor design. Both run at 66 MHz and offer real-time tracing, performance analysis, and 10-ns triggering.	83
Software			
Apple Computer	PlainTalk tool kit	Developer's tools create Macintosh-based applications that convert typed text to spoken English through concatenative Text-to-Speech technology, an extension of System 7. The tool kit contains Text-to-Speech Manager, various voices, and two engines: basic-level MacinTalk 2 and high-quality-voice MacinTalk Pro 2. A Macintosh Plus or greater is required. <i>\$1,000 to \$1,500 per year (worldwide redistribution license), depending on engine.</i>	84
Hippo Software	Hippix package	Set of commands and programming library allow Unix users to integrate OS/2 2.0 and Windows NT personal computers into their Unix networks. The command set implements most of IEEE Posix Std. 1003.2 and 1003.2a. The library supports over 90% of Posix 1003.1 API functions. <i>\$239; \$179 (commands only).</i>	85
Miscellaneous			
Telex Communications	MagnaByte M2X panel	Multimedia LCD projection system contains a built-in audio system with speaker and amplifier. Action! Windows/Macintosh presentation software, cables, and carrying case. PC users can plug in an interface bus to upgrade for Windows digital audio support. Users with Notebook and other portables can add audio to presentations without added hardware. <i>From \$5,195.</i>	86

Micro Review

continued from p. 5

cedures that you can find documentation for. One of the Visual Basic help files contains declarations for all of the DLL procedures in the Windows 3.1 application programming interface. You can simply press a button to copy them from the help file and paste them into your Basic program.

Visual Basic also lets you use the Windows object linking and embedding (OLE) feature to incorporate items like spreadsheets or graphics in your application. You can manipulate them from within your application or see the updated versions when they change in their original locations.

Once you get hooked on Windows

programming with Visual Basic, you can play to your heart's delight with its more advanced features. Some are as mundane as interfacing to a relational database using SQL. Others, like three-dimensional buttons or animated icons can exercise your creativity.

If you want to be able to throw together an ad hoc Windows application the way you used to do with Basic on your CPM system, this is the package for you.

Running Visual Basic 3.0 for Windows, Ross Nelson (Microsoft Press, Redmond Wash., 1993, 346 pp.; \$22.95)
Microsoft Visual Basic Workshop, Windows Edition, John Clark Craig (Microsoft Press, Redmond Wash., 1993, 504 pp. plus diskette; \$39.95)

These two books take different ap-

proaches to introducing you to Visual Basic. Nelson takes you systematically through the features, concocting ad hoc examples appropriate to each topic. Craig builds realistic applications, each emphasizing a particular feature, then explains each in detail. If you're serious, get them both. If not, get Nelson's book; it's shorter and cheaper.

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 183 Medium 184 High 185

**Late Magazines?
No Magazines?
Membership
Status Problems?
No Answers
To Your
Complaints?**

**Let your
Computer
Society
Ombudsman
cut
through
the red
tape
for you.**

Ombudsman
IEEE Computer Society
10662 Los Vaqueros Circle
PO Box 3014
Los Alamitos, CA
90720-1264
email: membership@
compmail.com
fax: (714) 821-4641



Classified Advertisement

RATES: \$10.00 per lines (ten lines minimum). Average five typeset words per line. Send copy at least one month prior to publication date to: Marian B. Tibayan, IEEE MICRO, 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1264; (714) 821-8380; fax (714) 821-4010. E-mail: m.tibayan@compmail.com.

SOUTH DAKOTA SCHOOL OF MINES & TECHNOLOGY New Programs in Computer Engineering

Department of Electrical Engineering, South Dakota School of Mines & Technology: Continuation of Faculty Search. Applications are invited for an Assistant or Associate Professor level, tenure track faculty position in the area of Computer Engineering. The school has a new program leading to a B.S. degree in Computer Engineering and the successful candidate will be involved in developing this new program. It is expected that the position will be filled by January 1, 1994, but the search will remain open until the position is filled. Duties will include developing and teaching undergraduate courses in the new Computer Engineering program, teaching undergraduate and graduate courses in the Electrical Engineering program, promoting and developing research, and directing research of graduate students. The areas of interest include all the fields of Com-

puter Engineering, especially those with a hardware oriented emphasis towards either VLSI design, microprocessors or digital systems. Applicants must possess a doctoral degree in Computer or Electrical Engineering, or be scheduled to complete all degree requirements, preferably by January 1, 1994. Salary is commensurate with qualifications and experience. South Dakota School of Mines and Technology, founded in 1885, has an enrollment of approximately 2,500 students and offers degrees in the major branches of engineering and the physical sciences. Applications must include a complete resume, indicating the actual or scheduled date of completion of all degree requirements, a statement of teaching and research interests, and names and addresses of three references. The applications should be sent to: Dr. A. L. Riemenschneider, Department Head, Electrical Engineering Department, South Dakota School of Mines and Technology, 501 East St. Joseph Street, Rapid City, SD 57701-3995, phone (605) 394-2451. Screening of applications will begin on November 1, 1993. South Dakota School of Mines and Technology does not discriminate on the basis of race, color, national origin, sex, religion, age or disability in employment or the provision of service.

Micro Law

continued from p. 3

models of legal protection. (Such recent decisions include those in the *Galoob*, *Sega*, and *Altai* cases.) These recent decisions have also reemphasized the principle of copyright law that predominantly functional and utilitarian aspects of works are not protected by copyright.

Model difficulties

The difficulties with the two principal models, patents and copyrights, center on the fact that they do too little—and do too much—for software and its noncode aspects. For example, in principle neither copyrights nor patents protect ideas. Yet, some of the most creative and valuable aspects of software may be classified as ideas. At least they are defined at so high a level of abstraction that they are, for purposes of patent law, unprotected ideas or mathematical principles. For purposes of copyright law, they are unprotected ideas, procedures, processes, systems, methods of operation, concepts, principles, or discoveries. Algorithms and user interfaces are salient examples. So, too, are languages and instruction sets. The patent and copyright models, unless they are greatly altered, do not allow for protection of highly abstract subject matter.

The fairly rigid requirement of the two traditional intellectual property legal models that injunctions must be issued against infringement creates problems, too, for the idea aspects of software. Congress and the courts have been faced with the choice of either ordering a total prohibition against any use by the defendant of the plaintiff's allegedly protected software idea or else making a determination that the allegedly infringing material is outside the scope of the plaintiff's statutory right. Both bodies have often preferred the underprotection of the second alternative to the overprotection of the first. Because the copyright and patent

models insist on doing too much for copyright and patent owners, in some ways, our present laws may give too little to the creators of new software or those who invest in bringing it to the market. On the other hand, the alternative under the two existing models would be, presumably, worse because it is excessive.

Indeed, the principal models of intellectual property protection not only at times do too little or do too much for the idea aspects of software, but they are often completely askew to software. Software and the intellectual property laws are often like ships passing in the night. Or perhaps you will prefer the metaphor of the Second and Ninth Circuits in the *Altai* and *Sega* cases: Using one of them for software is like "forcing a square peg into a round hole."

There are other legal models for industrial property protection, however. Copyright and patent law do not exhaust the known legal schemes for protection of intangible rights in intellectual creations.

Another model

Other legal models are known in other countries. One alternative to consider for industrial property protection of software is that of the utility model, known in many European countries and Japan. A utility model is a kind of petty patent. It typically has narrower scope, it may be allowable upon mere registration with a government agency, its validity is sustained on the basis of a lesser showing of technical merit than a regular patent, and it may have a shorter life span. The US Semiconductor Chip Protection Act of 1984 is basically a utility model law, and represents the first major departure in the US from the patent and copyright models of intellectual property law. Some of this law's provisions may suggest a more appropriate legal scheme for protection of computer software than the patent and copyright models do.

Using utility model laws as an alternative model for industrial property

protection suggests that some of the following features for a software protection law directed to noncode aspects of computer software and nonverbatim copying of computer programs may be appropriate. The following sketch of an industrial property law for protecting rights in future software technology presents a tentative model. I often use the word *probably*. Where I do not, you should understand it as implicit. This is a matter that calls for studying, discussing, and identifying the interests at stake, balancing them, and making appropriate trade-offs and a considerable number of other compromises. At this stage, it is more appropriate to raise questions, identify issues, and point to a possible model than it is to purport to prescribe definitive solutions.

First, something like utility model protection could be afforded to anything in the *index prohibitorum* of section 102(b) of the US Copyright Act. Any ideas, procedures, processes, systems, methods of operation, concepts, principles, or discoveries are excluded by copyright law from copyright protection. But Congress is free to protect them under a different law if that appears wise. By the same token, patent law's exclusion of laws of nature and mathematical principles (for example, algorithms) need not be a limitation on property rights in a utility model. Of course, common sense and considerations of policy may call for some of the same limits, but that would be a matter addressed to the wisdom of the legislature, not its power.

I assume utility model protection would be considered for algorithms, languages, instruction sets, icons, and some aspects of dataflow schemes and so-called sequence, structure, and organization of computer programs. More generally, utility model protection could be provided for any economically valuable aspects of future software technology for which Congress judged that industrial property protection is needed to encourage their creation, disclosure, and commercialization.

Protection of such aspects of computer software would probably be incompatible with adoption for software of the full scope of patent and copyright relief and remedies now in use. For example, injunctions against use of such software features by others may well be inconsistent with promotion of rapid software progress. Moreover, locking up new algorithms or instruction sets tightly for a period of years is probably a mistake.

It is one thing to make users pay reasonable compensation for the use of such new ideas, but quite another thing to permit their creators to withhold their use at will or levy any charge they please, with the aid of state compulsion. That means that a scheme for just and reasonable compensation—similar to that applied in the Federal Claims Court when the government uses or takes intellectual property rights belonging to others—would need to be considered.

As in the case of the semiconductor chip law and many utility model laws, an intermediate level of required technical merit would probably be appropriate. That would be one between patent law's inventive step and copyright law's *de minimis* standard. However, in-depth evaluation of the technical merit of particular software probably should be saved for litigation, if it ever occurs, as our semiconductor chip law provides.

A utility model law for software would probably call for registration of software rights, and immediate attachment of industrial property rights, while avoiding the high front-end costs of a patent-type examination procedure. The expense of a thorough analysis of the prior work in the field and of the advance over prior work embodied in the software feature in question would therefore be deferred until a plaintiff and a defendant are concerned enough about the matter to be willing to spend the kind of money involved in a lawsuit.

(There are possible arguments on the other side, however, as well. It may be

thought that the potential terrorism effect—in legal parlance, *in terrorem* effect—of intellectual property rights calls for advance screening of validity issues, so that would-be competitors are not scared off. I think, while others do not, that the social cost of legal terrorism is substantially exceeded by the social cost of having to make a thorough validity study of the 99.9 percent of software innovations that will never become commercially important and stir up litigation.)

The scope of software rights should probably be more like those of patent law than copyright law. Control over use is a major right omitted by copyright law, doubtless for good reasons in the case of copyright's traditional subject matter, but the right is economically important for software. Hence, software rights, like patent rights, should include control of use of the protected subject matter. Such uses would include, for example, execution of programs implementing protected algorithms or written or protected programming languages.

Certainty as to scope of industrial property rights is desirable. But the specificity of patent law's claims may be too difficult and expensive to emulate. On the other hand, the completely open-ended nature of copyright—what you see is what you get, and a court will eventually decide what that is—is also problematical. Some sort of intermediate compromise is in order, but I am unable to describe it for you. This aspect of the system needs more study. (So do most of the other features of the proposed model. This is a proposal for thinking about the issues, not an off-the-shelf solution.)

Independent creation of algorithms and other software features is another issue in search of a good solution. Like scope, it calls for deliberation and perhaps compromise. Patent and copyright laws have opposite answers to offer on this point. (As indicated earlier, independent creation of the infringing subject matter is an absolute defense

***Software rights,
like patent rights,
should include
control of use of
the protected
subject matter.***

to a claim of copyright infringement, and is no defense at all to a claim of patent infringement.) The US chip law was, unfortunately, silent on the issue.

Clearly, software utility model law will not draft itself. It is a substantial undertaking. But the expenditure of effort in undertaking study and drafting is worthwhile. It is a better course than trying to sweep the difficult issues under the rug by purporting to incorporate by reference some other existing body of law in its entirety. That is what the proponents of the 1984 Semiconductor Chip Protection Act initially tried to do, and in fact persuaded the Senate to endorse. (The Senate passed a chip bill that simply extended copyright law to cover chip layouts.) But the chair of the Intellectual Property Subcommittee in the House of Representatives (Robert Kastenmeier) wisely blocked that move. His opposition to a copyright incorporation-by-reference solution forced a choice of a so-called *sui generis* chip law. (A *sui generis* intellectual property law is one that does not fit under either traditional copyright law or traditional patent law. It is like copyright law in some respects, like patent law in some respects, and like neither of them in other respects.)

The *sui generis* chip law had to be specially drafted to address the specific perceived problems of the semiconductor industry. Unfortunately, drafting a sound software protection law is probably one or two orders of

magnitude more difficult than drafting a chip protection law.

Modifying existing laws

In any discussion of adopting other legal models for software, one may ask several questions. Why depart from the copyright and/or patent law models at all? Would it not be preferable just to modify one or both of these existing schemes to put some corners in the holes so that they will accommodate square pegs? Are they not familiar legal schemes with established, known bodies of case law helping us to fill in necessary omissions or ambiguities in any statute applied to software? And why dispense with the international recognition of intellectual property rights classified as copyrights or patents, and thus forego automatic transnational protection of software rights under international treaties?

Congress answered many of these questions in the *House Report on the Semiconductor Chip Protection Act of 1984*. The House refused to follow the Senate's copyright approach to chip protection and insisted on enactment of a *sui generis* chip law instead (H. Rep. No. 98-781, 98th Cong., 2d Sess. 7-8, 1984). It recognized that, to a significant extent, the Berne Convention and the existing body of US copyright and patent law do not permit us to square off the legal hole to make the software peg fit it. (For example, the Berne Convention does not permit reduction of the term of copyright protection substantially below its present 50- to 75-year term. Nor does it permit a copyright law to "discriminate" against software, literary-work copyrights by denying them injunctions and, in effect, opening them up to the compulsion of unconsented-to use. Protection of abstract ideas and systems is infeasible under our existing intellectual property laws.)

Indeed, to modify copyright law and patent law sufficiently so that they accommodate software properly would mean changing them so much that they

would no longer be copyright or patent law. Not only would that be infeasible, but it would be unacceptable to the users and beneficiaries of traditional copyright and patent law.

International protection is speculative, apart from verbatim copying. There is no automatic transnational copyright protection, apart from verbatim or near-verbatim copying of programs. There is no worldwide consensus in favor of protecting nonliteral aspects of software under copyright law. Indeed, in the wake of the *Altai* and *Sega* decisions, there may no longer be such domestic protection.

Moreover, if the US tried to protect ideas and other abstract or nonliteral aspects of software under one or both of the two principal existing systems, patents and copyrights, serious trade problems would occur. US treaty obligations would then oblige us to protect such aspects of foreign nationals' software while their home countries continued to refuse to protect the same aspects of US companies' software. That is required by the principle of equal "national treatment," the cornerstone of our intellectual property treaties. (See Berne Convention, art. 5(1); Paris Convention, art. 2(1).) As one scholar has pointed out: "A unilateral broadening of protection at home can thus weaken a [treaty] Member State's overall competitive position."

Finally, the concept of saving effort by relying on the existing body of copyright law is an illusion. Earlier, I gave a few examples of difficult choices that may need to be made, in sketching the outline of a software law based on the legal model of a utility model. There are many more difficult choices. They should be addressed explicitly, not by pretending that some existing body of law that has never addressed, or had any reason to address, such issues offers a ready-made solution to be incorporated by reference. The likelihood of coming to rational solutions for these problems is more enhanced by addressing them explicitly, than by hoping that buying a lottery ticket—

meaning incorporating a preexisting law by reference—will lead to a serendipitous outcome. (Murphy's Law is inconsistent with your getting free lunches.) The right way to craft an appropriate scheme of industrial property protection for software is to do it purposefully.

What's next?

The most recent trend of decision in software copyright cases is away from *Whelan*, away from protection of nonliteral aspects of computer programs, and away from treating copyrights as if they were patents. That does not mean that industrial property protection of nonliteral, relatively abstract aspects of software is a bad idea. Nor does it mean that courts should or do consider these aspects of software to be without economic value and undeserving of any kind of industrial property protection. It means only that the traditional intellectual property law models, particularly that of copyright law, do not accommodate such protection. We need to consider another model. That in turn would mean that legislation of some kind is needed to accord such protection.

Before legislation is attempted, however, much more study and thought are needed. Legislation should only follow both study and an attempt to build an informed consensus among software professionals. That implies developing a consensus in crafting, and then in support of, an appropriate law for industrial property protection of the valuable aspects of software technology. How could or should that be done?

Congress attempted to address this question in the 1970s. It established a National Commission on New Technological Uses of Copyrighted Works (CONTU) to study and advise Congress what to do about protecting computer software. CONTU presented its final report to Congress in 1979. CONTU's majority (over a strong dissent) punted in favor of letting the status quo develop further, under the copyright law, and Congress accepted that recommen-

dation. That resulted in the last decade or so of total confusion and legal mayhem in the software protection field.

Efforts since then within the IEEE Computer Society's Committee on Public Policy (COPP) and IEEE-USA's Intellectual Property Committee have not been notably successful. Perhaps, the IEEE is not institutionally well suited for this kind of process. The IEEE seems to work better with reactive projects than proactive projects.

Possibly, academia is more well suited to nurture the proposed computer software law project. But then there is the issue of overcoming distance (or at least perceived distance) from reality. My present view is that a university setting is more suited to this process (at least initially) than any other. But academia needs a very large amount of nonacademic leavening to accomplish satisfactory, realistic results.

By the time this issue of *IEEE Micro* reaches its readers, one preliminary effort in this general direction will have occurred. On October 1, 1993, the George Washington University's School of Engineering and Applied Science (GWU SEAS), in cooperation with the University of Wisconsin's Kastenmeier Foundation, is hosting a one-day Technology Workshop on Computer Software Protection. The workshop chair is Gideon Frieder, dean of GWU SEAS, who has long been interested and involved in computer software/intellectual property law problems.

I have assisted the dean in this project for several reasons. One is that I am the person who teaches computer software copyright and patent law at GWU. Moreover, I have been concerned professionally about these issues for 25 years. I have become more and more concerned—and indeed apprehensive—that the US may be going in the wrong direction in this field, with a likely adverse consequence, among others, of loss of its world leadership of computer software progress. The time has come to substitute positive action for mere concern.

The workshop participants are a mix-

ture of academicians (mainly EE and CS Department), computer industry business representatives, other computer science professionals, and government representatives, along with a minimal sprinkling of intellectual property lawyers. The plan is to describe and discuss computer software technology of the future (1990s-2010). At the outset, we need to ascertain whether and to what extent, if any, computer software evolution will strain present intellectual property law; consider possible effects on software progress and the public interest; and consider whether another legal model for computer software protection is needed, or at least needs to be considered.

The purpose of the workshop is more to ventilate views and open the door to future discussions and study than to reach firm conclusions. Some workshop participants are known to favor the legal status quo, and to feel that we only need to give present copyright law (and patent law) more time and opportunity to evolve and develop. These participants favor evolution of computer software protection by the judicial route—let the courts work out proper software protection, on a case-by-case basis.

Others have different views; readers of this column will know that I am among those skeptical about the likelihood of success of that course of action, and believe in legislation by Congress rather than by the courts. The organizers of this workshop consider it more important simply to discuss the issues and explore the range of opinions than to determine which views are right and which wrong—if that is even a meaningful notion in this context.

Further workshops are planned to follow up on various issues, possibly over a span of several years. As the earlier part of this column suggests, it is unclear what the best answers are to many questions about the model and architecture. We hope a more informed view will result from further study and discussion, and that a clearer description of the appropriate model for legally protecting computer software will

***It is more
important simply
to discuss the
issues and
explore the range
of opinions than
to determine
which views are
right and which
wrong.***

emerge. Ideally, the train of discourse will end with one of two proposals. We could propose a federal computer software protection law for algorithms, instruction sets, programming languages, and other nonliteral aspects of computer software technology, based on an improved legal model. Or, we could abandon the whole project as quixotic or infeasible. Perhaps there are other possibilities.

Readers who would like to participate actively in this project in 1994 can write Gideon Frieder, School of Engineering and Applied Science, George Washington University, 725 23rd St. NW, Washington, DC 20052; frieder@seas.gwu.edu. You can also write to me at the address shown on the first page of this column.

Reader Interest Survey

Indicate your interest in this department by circling the appropriate number on the Reader Service Card.

Low 177 Medium 178 High 179

Advertiser/Product Index

FOR DISPLAY ADVERTISING INFORMATION, CONTACT:

Southern California and Mountain States: Richard C. Faust, Douglas C. Faust, 24050 Madison Street, Suite 101, Torrance, California 90505; Tel: (310) 373-9604; Fax: (310) 373-8760.

Northern California and Pacific NW: William W. Hague, 9017 Peacock Hill Road, Gig Harbor, Washington 98332; Tel: (206) 858-7575; Fax (206) 858-7576.

East Coast: Gail A. Frank, Nancy Inserra, 82 Bethany Road, Suite 4, Hazlet, New Jersey 07730; Tel: (908) 264-1100; Fax: (908) 264-4340.

New England: Paul Gillespie, PO Box 6444, Holliston, Massachusetts 01746; Tel: (508) 429-8907; Fax: (508) 429-8684.

Southwest: Frank E. Johnson, 3601 Smith-Barry Road, Suite 103, Arlington, Texas 76013; Tel: (817) 275-2651; Metro Voice/Fax: (817) 265-3811.

Midwest: Harold L. Leddy, 345 Auburn Avenue, Winnetka, Illinois 60093-3603; Tel: (708) 446-8764; Fax: (708) 446-7985.

Advertising Manager: Heidi Rex, 10662 Los Vaqueros Circle, Los Alamitos, California 90720-1264; Tel: (714) 821-8380; Fax: (714) 821-4010.

For production information, conference, and classified advertising, contact Marian Tibayan. *IEEE MICRO*, 10662 Los Vaqueros Circle, PO Box 3014, Los Alamitos, California 90720-1264; Tel: (714) 821-8380; Fax: (714) 821-4010; email: m.tibayan@compmail.com.

Moving?

**PLEASE NOTIFY
US FOUR WEEKS
IN ADVANCE**

Name (Please Print)

New Address

City

State/Country

Zip

Mail to:
IEEE Computer Society
Circulation Department
PO Box 3014
10662 Los Vaqueros Circle
Los Alamitos, CA 90720-1264

**ATTACH
LABEL
HERE**

- This notice of address change will apply to all IEEE publications to which you subscribe.
- List new address above.
- If you have a question about your subscription, place label here and clip this form to your letter.

	RS #	Page #
Analogic	28	95
Apple Computer	84	98
AST Research	25	95
AT&T	10	93
Aurora Technologies	17	94
Belkin Components	21	95
Bit 3 Computer	80	98
CNet Technology	12	93
Data Translation	30	96
Delta Computer Systems	81	98
DWB Associates	16	94
Eo, Inc.	10	93
Flomerics	33	96
Frederick Engineering	14	94
Hippo Software	85	98
IEEE CS Ombudsman	—	99
IEEE CS Press	—	23, 48, 68, 78, 89
Kluwer Academic Publishers	1	C.IV
Knights Technology	36	97
Mentor Graphics	37	97
Mesa Electronics	31	96
MicroTouch Systems, Inc.	24	95
Microway	82	98
Mitsubishi Electronics America	35	97
MUX Lab	20	94
Pacific Cyber/Metrix	27	95
Planar Systems Inc.	23	95
Raytheon	38	97
SBE	19	94
Sharp Electronics	29	96
Silconix	22	95
Smart Modular Technologies	11	93
Spectrum Signal Processing	26	95
Talking Technology	15	94
Tanner Research	32	96
Tektronix	83	98
Telex Communications	86	98
USAR Systems	18	94
Viewlogic	34	96
VLSI Technology	13	93
Classified Advertising	—	99

THE FOLLOWING INFORMATION IS AVAILABLE:

Contact the Publications Office;
to facilitate handling, please request by number.

- Membership application, student #203, others #202
- Publications catalog #201
- Technical committee list/application #197
- Chapters lists, start-up procedures #193
- Student scholarship information #192
- Volunteer leaders/staff directory #196
- IEEE senior member grade application #204
(requires ten years practice and significant performance in five of those ten)

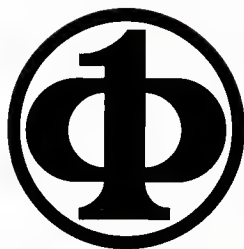
To check membership status or report a change of address, call the IEEE toll-free number, 1-800-678-4333. Direct all other Computer Society-related questions to the Publications Office.

PURPOSE

The IEEE Computer Society advances the theory and practice of computer science and engineering, promotes the exchange of technical information among 100,000 members worldwide, and provides a wide range of services to members and nonmembers.

MEMBERSHIP

Members receive the acclaimed monthly magazine *Computer*, discounts, and opportunities to serve (all activities are led by volunteer members). Membership is open to all IEEE members, affiliate society members, and others interested in the computer field.



IEEE COMPUTER SOCIETY

A member society of the
Institute of Electrical and Electronics Engineers, Inc.

PUBLICATIONS AND ACTIVITIES

Computer. An authoritative, easy-to-read magazine containing tutorial and in-depth articles on topics across the computer field, plus news, conferences, calendar, interviews, and product reviews.

Periodicals. The society publishes eight magazines and seven research transactions. Refer to membership application or request information as noted at left.

Conference Proceedings, Tutorial Texts, Standards Documents. The Computer Society Press publishes more than 100 titles every year.

Standards Working Groups. More than 100 of these groups produce IEEE standards used throughout the industrial world.

Technical Committees. More than 30 TCs publish newsletters, provide interaction with peers in specialty areas, and directly influence standards, conferences, and education.

Conferences/Education. The society holds about 100 conferences each year and sponsors many educational activities, including computing science accreditation.

Chapters. Regular and student chapters worldwide provide the opportunity to interact with colleagues, hear technical experts, and serve the local professional community.

OMBUDSMAN

Members experiencing problems — magazine delivery, membership status, or unresolved complaints — may write to the ombudsman at the Publications Office.

EXECUTIVE COMMITTEE

President: James H. Aylor*
University of Virginia
Thornton Hall/Electrical Engineering
Charlottesville, VA 22903
Phone: (804) 924-6100
Fax: (804) 924-8818
E-mail: jha@virginia.edu

President-Elect: Laurel V. Kaleda*
Past President: Bruce D. Shriver*

VP, Technical Activities: Joseph Boykin (1st VP)*
VP, Conferences and Tutorials: Anneliese Von Mayrhauser (2nd VP)*
VP, Educational Activities: Gerald L. Engel†
VP, Membership Activities: Fiorenza C. Albert-Howard*
VP, Press Activities: Ronald G. Hoelzeman†
VP, Publications: Barry W. Johnson†
VP, Standards Activities: Gary S. Robinson†

Secretary: Mario R. Barbacci*
Treasurer: Michael Evangelist*
IEEE Division V Director: Bill D. Carroll†
IEEE Division VIII Director: V. Tom Rhyne†
Executive Director: T. Michael Elliott*

*voting member of the Board of Governors
†nonvoting member of the Board of Governors

BOARD OF GOVERNORS

Term Expiring 1993:

Fiorenza C. Albert-Howard, Jon T. Butler,
Michael C. Mulder, Yale N. Patt, Benjamin W. Wah,
Ronald Waxman, Akihiko Yamada

Term Expiring 1994:

Mario R. Barbacci, L. Felipe Cabrera, Wolfgang K. Giloi, Guylaine M. Pollock, John P. Riganati, Ronald D. Williams, Thomas W. Williams

Term Expiring 1995:

Fletcher J. Buckley, Doris L. Carver, Elliot J. Chikofsky,
Joanne E. DeGroat, Michael J. Flynn,
Mary Jane Irwin, Grace C.N. Wei

Next Board Meeting

November 12, 1993, 8:30 a.m.
Santa Clara Marriott, Santa Clara, Calif.

SENIOR STAFF

Executive Director: T. Michael Elliott
Publisher: H. True Seaborn
Director, Conferences and Tutorials: Anne Marie Kelly
Director, Finance and Information Services: Deborah Rafal
Director, Board and Administrative Services: Violet S. Doan
Assistant to the Executive Director: Sandra K. Plau

COMPUTER SOCIETY OFFICES

Headquarters Office

1730 Massachusetts Ave. NW
Washington, DC 20036-1903
Phone: (202) 371-0101
Fax: (202) 728-9614
E-mail: hq.ofc@compmail.com

Publications Office

10662 Los Vaqueros Cir.
PO Box 3014
Los Alamitos, CA 90720-1264
Membership and General Information:
(714) B21B3B0
membership@compmail.com
Publication Orders: (800) 272-6657
Fax: (714) B21-4010
E-mail: cs.books@compmail.com

European Office

13, Ave. de L'Aquilon
B-1200 Brussels, Belgium
Phone: 32 (2) 770-21-98
Fax: 32 (2) 770-B5-05
E-mail: euro.ofc@compmail.com

Asia/Pacific Office

Doshima Building
2-19-1 Minami-Aoyama, Minato-ku
Tokyo 107, Japan
Phone: B1 (3) 340B-3118
Fax: B1 (3) 340B-3553
E-mail: tokyo.ofc@compmail.com



IEEE OFFICERS

President: Martha Sloan	Secretary: Souguil J.M. Ann	VP, Educational Activities: Edward A. Parrish	VP, Regional Activities: Luis T. Gandia
President-Elect: H. Troy Nagle	Treasurer: Theodore W. Hissey, Jr	VP, Professional Activities: Charles K. Alexander	VP, Standards Activities: Wallace S. Read
Past President: Merrill W. Buckley, Jr.		VP, Publication Activities: Helen M. Wood	VP, Technical Activities: Donald M. Bolle

REAL-TIME SYSTEMS

The International Journal of Time-Critical Computing Systems

Expanding to six issues per year in 1994

John A. Stankovic, *University of Massachusetts, Amherst, USA*

Wolfgang A. Halang, *Fern Universität Hagen, Germany*

Marlo Tokoro, *Keio University, Japan*

Volume 5, issue 1, March 1993

Special Issue: Real-Time Languages and Language-Level Timing Tools and Analysis

Guest Editor: Alexander D. Stoyenko

Editorial: Real-Time Kernel Interfaces, John A. Stankovic

A Partial Evaluator for the Maruti Hard Real-Time System,

Vivek Nirkhe and William Pugh

Predicting Program Execution Times by Analyzing Static and Dynamic

Program Paths, Chang Yun Park

RTC: Language Support for Real-Time Concurrency,

Victor Fay Wolfe, Susan Davidson, and Insup Lee

Timing Analysis of MRL: A Real-Time Rule-Based System,

Chih-Kan Wang and Aloysius K. Mok

Volume 5, issues 2/3, May 1993

Special Issue: Incremental Prototyping Technology for Embedded Real-Time Systems

Guest Editor: Sandro Bologna

Editorial: Resource Allocation in Real-Time Systems, John A. Stankovic

Guest Introduction: The IPTES Project, Sandro Bologna

IPTES: A Concurrent Engineering Approach for Real-Time Software Development, P. Pulli and R. Elmstrom

The IPTES Environment: Support for Incremental Heterogeneous and Distributed Prototyping, Gonzalo León, Juan Carlos Dueñas, Juan

A. de la Puente, Nabli Zakhamas, and Alejandro Alonso

Graphical Animation as a Form of Prototyping Real-Time

Software Systems, P. Pulli, M. Heikkinen and R. Lintulampi

An Executable Subset of VDM-SL, in a SA/RT Framework,

René Elmström, Poul Bøgh Lassen and Michael Andersen

Distributed Execution of Specifications, Juan A. de la Puente,

Alejandro Alonso, Gonzalo León and Juan Carlos Dueñas

High-Level Timed Petri Nets as a Kernel for Executable Specifications,

Miguel Felder, Carlo Ghezzi and Mauro Pezzé

Giving Semantics to SA/RT by Means of High-Level Timed Petri Nets,

René Elmström, Ralno Lintulampi and Mauro Pezzé

Volume 5, issue 4, October 1993

A Formalization of Priority Inversion,

Özalp Babaoglu, Keith Marzullo, and Fred B. Schneider

Deadlock Prevention in Concurrent Real-Time Systems,

Susan Davidson, Insup Lee and Victor Fay Wolfe

Pipelined Processors and Worst Case Execution Times,

N. Zhang, A. Burns, and M. Nicholson

A Comparison of Four Microcomputer Operating Systems,

George Wells

SUBSCRIPTION INFORMATION

(ISSN 0922-6443) 1993, Volume 5 (4 issues)

Institutional Rate: \$212.00/Dfl. 392.00

Individual Rate: \$85.00/ Dfl. 205.00

A PRACTITIONER'S HANDBOOK FOR REAL-TIME ANALYSIS:

Guide to Rate Monotonic Analysis for Real-Time Systems

Carnegie Mellon University/Software Engineering Institute

by Mark Klein, Thomas Ralya, Bill Pollak, Ray Obenza,

and Michael González Harbour

The *Handbook* contains an invaluable collection of quantitative methods that enable real-time system developers to understand, analyze, and predict the timing behavior of many real-time systems. The methods are practical and theoretically sound, and can be used to assess design tradeoffs and troubleshoot system timing behavior. This collection of methods is called *rate monotonic analysis (RMA)*.

This *essential* guide includes a framework for describing and categorizing the timing aspects of real-time systems, step-by-step techniques for performing timing analysis, numerous examples of real-time situations to which the techniques can be applied, and two case studies.

A Practitioner's Handbook for Real-Time Analysis: Guide to Rate Monotonic Analysis for Real-Time Systems has been created to serve as a definitive source of information and a guide for developers as they analyze and design real-time systems using RMA. The *Handbook* is an excellent reference, and may be used as the text for advanced courses on the subject.

1993 ISBN 0-7923-9361-9

712 pp.

\$95.00/ Dfl. 195.00

Kluwer Real-Time Titles

The Testability of Distributed Real-Time Systems

By Werner Schütz

1993 ISBN 0-7923-9386-4 160 pp. \$69.95/ Dfl. 145.00

Formal Techniques in Real-Time and Fault-Tolerant Systems

Edited by Jan Vytöpil

1993 ISBN 0-7923-9332-5 224 pp. \$79.00/ Dfl. 160.00

Synchronous Programming of Reactive Systems

By Nicolas Halbwachs

1993 ISBN 0-7923-9311-2 184 pp. \$79.95/ Dfl. 157.50

Real Time Systems Engineering and Applications

Edited by Michael Schlebe and Saskia Pferrer

1992 ISBN 0-7923-9196-9 464 pp. \$102.50/ Dfl. 200.00

Real-Time UNIX® Systems Design and Application Guide

By Borko Furht, Dan Grostlick, David Gluch, Guy Rabbat, John Parker, Meg McRoberts

1991 ISBN 0-7923-9099-7 352 pp. \$63.50/ Dfl. 133.00

Foundations of Real-Time Computing: Formal Specifications and Methods

Edited by André M. van Tilborg and Gary M. Koob

1991 ISBN 0-7923-9167-5 336 pp. \$59.95/ Dfl. 135.00

Foundations of Real-Time Computing: Scheduling & Resource Management

Edited by André M. van Tilborg and Gary M. Koob

1991 ISBN 0-7923-9166-7 336 pp. \$59.95/ Dfl. 130.00

Synchronization in Real-Time Systems

By Rangunathan Rajkumar

1991 ISBN 0-7923-9211-6 208 pp. \$59.95/ Dfl. 140.00

Constructing Predictable Real Time Systems

By Wolfgang A. Halang and Alexander D. Stoyenko

1991 ISBN 0-7923-9202-7 352 pp. \$75.00/ Dfl. 170.00

